



Márcio Borges Medeiros

Licenciatura em Engenharia Informática

Identificação de ninhos de cegonha em postes eléctricos

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática

Orientador: Francisco Azevedo, Professor Auxiliar, FCT/UNL

Júri:

Presidente: Prof. João Manuel dos Santos Lourenço

Arguentes: Prof. Paulo Lobato Correia

Vogais: Prof. Francisco Azevedo



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Junho, 2013

Identificação de ninhos de cegonha em postes eléctricos

Copyright © Márcio Borges Medeiros, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Agradecimentos

Começo por agradecer ao meu orientador , Francisco Azevedo, pela paciência e por todo o apoio demonstrado ao longo dos últimos meses. A sua ajuda foi fundamental para a conclusão desta dissertação. Queria também deixar o meu agradecimento ao professor Nuno Correia, que me aconselhou numa fase inicial, a uma solução ao problema apresentado nesta dissertação.

Gostaria também de agradecer à empresa Albatroz Engenharia por me proporcionar esta oportunidade. Em particular deixo a minha gratidão a João Gomes Mota, por todos os seus esclarecimentos e por disponibilizar o material de suporte a esta dissertação.

Por fim um agradecimento especial a toda a minha família e amigos, que sempre me apoiaram, mesmo nos momentos mais difíceis. Não podia terminar sem reconhecer o contributo dos meus colegas, em particular Bruno Santos, Renil Lacmane e Jorge Pereira. Um muito obrigado a todos.

Resumo

Incidentes envolvendo cegonhas representam a segunda maior causa para interrupções no fornecimento energético. As cegonhas encontram nos postes eléctricos um bom habitat para a construção dos seus ninhos. No entanto existem locais onde a sua existência não é desejável, tornando-se necessário a realização de vistorias dedicadas e se necessário, acções que visam a realocação dos ninhos.

Nesta dissertação propõe-se a criação de uma ferramenta capaz de automaticamente identificar ninhos de cegonha em postes eléctricos. Utilizando vídeos e/ou imagens capturadas para o efeito, são extraídas as arestas nas mesmas com recurso ao *Canny edge detector* e transformada de *Hough*. Verificandose a existência de um poste prossegue-se com a detecção dos ninhos de cegonha. Para este efeito foi construído um classificador em cascata baseado em *haar features*.

Nos testes efectuados foi possível identificar correctamente 91% dos postes eléctricos, existindo uma taxa de falsos positivos de 5,2%. Procedendo-se ao reconhecimento dos ninhos de cegonha, 81,3% foram correctamente detectados, contrabalançando com os 12,5% de detecções incorrectas.

Palavras-chave: Detecção de ninhos de cegonha, identificação de postes eléctricos, extracção de arestas, *Canny edge detector*, transformada de *Hough*, *haar features*.

Abstract

Incidents involving storks represent the second leading cause for interruptions in energy supply. Electric poles provide a good habitat for stork nests. However there are locations where their existence is not desirable, making it necessary to carry on monitoring activities and if required, relocation of problematic nests.

This thesis proposes the creation of a tool capable of identifying stork nests on electric poles. Using videos or images captured for this purpose, edges are extracted with the aid of Canny edge detector and Hough transform. Once the existence of the pylon is confirmed, we proceed with the stork nests detection. To achieve this goal, a cascade classifier based on haar features was built.

In the carried out tests, 91% of the pylons were correctly identified, with a false positive rate of 5,2%, Proceeding with the stork nest recognition, 81% were correctly detected, counterbalanced with 12,5% of incorrect detections.

Keywords: Stork nests detection, pylon detection, edge extraction, Canny edge detector, Hough transform, haar features.

Conteúdo

Introdução	1
1.1 Contexto.....	1
1.2 Descrição do problema	4
1.3 Objecto de Estudo.....	5
1.3.1 Cegonha Branca.....	5
1.3.2 Postes eléctricos.....	5
1.4 Proposta de Solução	6
1.4.1 Identificação de postes eléctricos.....	6
1.4.2 Identificação de ninhos de cegonha.....	7
1.5 Contribuições	8
1.6 Estrutura	9
Conceitos Fundamentais	11
2.1 Imagens de rastreio e imagens vectoriais	11
2.2 Filtros de Imagem	13
2.2.1 Filtros Lineares.....	13
2.2.2 Filtros não lineares	14
2.3 Canny Edge Detector.....	14
2.4 Hough Transform	17
2.5 Haar like features	19
2.5.1 Integral Images.....	21
2.5.2 AdaBoost	22
2.5.3 Cascade Classifier	25
2.6 OpenCV.....	25
Trabalho Relacionado.....	27

3.1	Identificação e Extracção de arestas.....	27
3.2	Detecção de objectos utilizando <i>Haar-features</i>	30
3.3	Identificação de postes eléctricos	35
3.4	Protótipos existentes.....	38
Solução Proposta		45
4.1	Visão geral.....	45
4.2	Análise de ficheiros vídeo.....	47
4.3	Pré Processamento de Imagens.....	49
4.4	Identificação de postes eléctricos	53
4.5	Detecção de ninhos de cegonha	60
Análise dos Resultados.....		67
5.1	Identificação de postes eléctricos em imagens	67
5.2	Identificação de postes eléctricos em vídeos.....	70
5.3	Comparação de resultados com outros trabalhos	71
5.4	Identificação de ninhos de cegonha em imagens.....	71
5.5	Identificação de ninhos de cegonha em vídeos.....	74
5.6	Comparação de resultados com outros trabalhos	75
Conclusão e Trabalho Futuro		77
Bibliografia.....		79
Anexo.....		85

Lista de Figuras

FIGURA 1. 1 - EVOLUÇÃO DO NÚMERO DE NINHOS DE CEGONHAS CONTABILIZADOS NOS RECENSEAMENTOS REALIZADOS EM PORTUGAL.	2
FIGURA 1. 2 - EVOLUÇÃO DA TAXA DE INCIDENTES CAUSADAS POR CEGONHAS NAS DÉCADAS DE 90' E 00'.....	3
FIGURA 1. 3 - VARIANTE DE POSTE MAT UTILIZADO NESTA TESE.	6
FIGURA 2. 1 - UTILIZAÇÃO DE UM FILTRO GAUSSIANO DE MATRIZ 11x11.....	15
FIGURA 2. 2 - RESULTADO DAS ARESTAS EXTRAÍDAS PELO <i>CANNY</i> APLICANDO DIFERENTES <i>THRESHOLDS</i>	17
FIGURA 2. 3 - IDENTIFICAÇÃO DA RECTA QUE INTERSECTA DOIS PONTOS NO ESPAÇO DE <i>HOUGH</i>	18
FIGURA 2. 4 - REPRESENTAÇÃO DO ESPAÇO DE <i>HOUGH</i> EM COORDENADAS POLARES.	19
FIGURA 2. 5 - CONJUNTO DE ALGUMAS <i>FEATURES</i> ADOPTADAS POR VIOLA E JONES.	20
FIGURA 2. 6 - REPRESENTAÇÃO DE UMA <i>INTEGRAL IMAGE</i>	21
FIGURA 2. 7 - FUNCIONAMENTO DO ADABOOST.	22
FIGURA 2. 8 - PROCESSO DE TREINO DO CLASSIFICADOR.....	24
FIGURA 2. 9 - FUNCIONAMENTO DO CLASSIFICADOR EM 'CASCATA'.....	25
FIGURA 3. 1 - IDENTIFICAÇÃO DA MATRÍCULA DE UM AUTOMÓVEL.....	29
FIGURA 3. 2 - RESULTADOS OBTIDOS POR VIOLA E JONES NO RECONHECIMENTO FACIAL.....	31
FIGURA 3. 3 - RESULTADOS DA DETECÇÃO LATERAL E TRASEIRA DE VEÍCULOS UTILIZANDO DIFERENTES CLASSIFICADORES.	34
FIGURA 3. 4 - CURVA ROC DO SISTEMA DE IDENTIFICAÇÃO DE PEDESTRES.	35
FIGURA 3. 5 - MODELO DE POSTE UTILIZADO POR GOLIGHTLY <i>ET AL</i>	36
FIGURA 3. 6 - RESULTADOS DO ALGORITMO DE DETECÇÃO DE POSTES ELÉTRICOS PROPOSTO POR TILAWAT <i>ET AL</i>	37
FIGURA 3. 7 - IDENTIFICAÇÃO DAS REGIÕES DO POSTE ONDE PODERÃO EXISTIR NINHOS.....	38
FIGURA 3. 8 - RESULTADOS DA IDENTIFICAÇÃO DE POSTES ELÉTRICOS POR JOÃO FERNANDES.....	42
FIGURA 3. 9 - RESULTADOS DA DETECÇÃO DE NINHOS DE CEGONHA POR JOÃO FERNANDES.....	43

FIGURA 4. 1 - VISÃO GERAL DO FUNCIONAMENTO DO ALGORITMO.....	46
FIGURA 4. 2 – VARIAÇÃO DO TEMPO (EM SEGUNDOS) NECESSÁRIO PARA O PROCESSO DE IDENTIFICAÇÃO EM FUNÇÃO DA RESOLUÇÃO DA IMAGEM.	50
FIGURA 4. 3 - RESULTADOS DA DETECÇÃO DE NINHOS UTILIZANDO IMAGENS DE DIFERENTES RESOLUÇÕES.	50
FIGURA 4. 4 - IMAGENS RESULTANTES APÓS TRANSFORMAÇÕES.....	51
FIGURA 4. 5 - RESULTADOS DA IDENTIFICAÇÃO DE POSTES DOS 4 MÉTODOS PROPOSTOS.....	52
FIGURA 4. 6 - ESQUEMATIZAÇÃO DO PRÉ-PROCESSAMENTO EFECTUADO.	53
FIGURA 4. 7 - ENCADEAMENTO DAS DIFERENTES OPERAÇÕES EFECTUADAS NA IDENTIFICAÇÃO DE UM POSTE.	54
FIGURA 4. 8 - ARESTAS IDENTIFICADAS PELO <i>CANNY EDGE DETECTOR</i> UTILIZANDO DIFERENTES <i>THRESHOLDS</i>	56
FIGURA 4. 9 - CONJUNTO DE RECTAS EXTRAÍDAS PELA TRANSFORMADA DE <i>HOUGH</i>	56
FIGURA 4. 10 - IDENTIFICAÇÃO DAS SECÇÕES DO POSTE CONSOANTE OS ÂNGULOS DAS SUAS ARESTAS.....	57
FIGURA 4. 11 - DIMENSÕES E PONTOS DE INTERESSE DO POSTE.	59
FIGURA 4. 12 - RESULTADO DO PROCESSO DE IDENTIFICAÇÃO DAS ARESTAS DO POSTE.....	60
FIGURA 4. 13 - EXCERTO DE ALGUMAS IMAGENS UTILIZADAS NO TREINO DO CLASSIFICADOR DE NINHOS.....	61
FIGURA 4. 14 - PRIMEIRA ETAPA DO PROCESSO DE TREINO DO CLASSIFICADOR.	62
FIGURA 4. 15 - RESULTADOS DA DETECÇÃO DE NINHOS DE CEGONHA UTILIZANDO DIFERENTES FACTORES DE ESCALA.	64
FIGURA 4. 16 - RESULTADOS DA DETECÇÃO DE NINHOS UTILIZANDO DIFERENTES VALORES PARA O PARÂMETRO <i>MINIMUM HIT</i>	65
FIGURA 4. 17 - DIAGRAMA DO PROCESSO DE DETECÇÃO DE NINHOS DE CEGONHA.....	66
 FIGURA 5. 1 - INCORRECTA IDENTIFICAÇÃO DE UM POSTE.....	 68
FIGURA 5. 2 - RESULTADOS DA DETECÇÃO DE NINHOS DE CEGONHA EM IMAGENS.	72
FIGURA 5. 3 - INCORRECTA IDENTIFICAÇÃO DE UM NINHO CONTIDO NA PAISAGEM.	73
 FIGURA 8. 1 - POSTE CORRECTAMENTE IDENTIFICADO	 85
FIGURA 8. 2 - INCORRECTA IDENTIFICAÇÃO DE UM POSTE.....	86
FIGURA 8. 3 - IDENTIFICAÇÃO DE UM NINHO NA PAISAGEM (FALSO POSITIVO).....	86
FIGURA 8. 4 - INCORRECTA IDENTIFICAÇÃO DE UM NINHO NO POSTE (FALSO POSITIVO).....	87
FIGURA 8. 5 - IDENTIFICAÇÃO DA TOTALIDADE DOS NINHOS EXISTENTES NO POSTE.	87
FIGURA 8. 6 - NÃO DETECÇÃO DE UM NINHO, ALIADO A DUAS IDENTIFICAÇÕES INCORRECTAS	88
FIGURA 8. 7 - IDENTIFICAÇÃO DE UM POSTE EM POSIÇÃO NÃO FRONTAL.	88
FIGURA 8. 8 - IDENTIFICAÇÃO DE 7 DOS 8 NINHOS EXISTENTES.	89
FIGURA 8. 9 - EXEMPLO DE UM POSTE CONTENDO UM GRANDE NÚMERO DE NINHOS (25).	89
FIGURA 8. 10 - SEQUÊNCIA DE ANÁLISE DAS <i>FRAMES</i> EXTRAÍDAS DE UM FICHEIRO VÍDEO.....	90

Lista de Tabelas

TABELA 1. 1 - LOCALIZAÇÃO DOS NINHOS EM POSTES MAT.	3
TABELA 3. 1 - RESULTADOS DA IDENTIFICAÇÃO DE VEÍCULOS.	28
TABELA 3. 2 - RESULTADOS DA IDENTIFICAÇÃO FACIAL DO ALGORITMO DE KE E KANG	31
TABELA 3. 3 - RESULTADOS DA IDENTIFICAÇÃO DA REGIÃO OCULAR DO ALGORITMO DE KE E KANG	32
TABELA 3. 4 - COMPARAÇÃO DOS RESULTADOS OBTIDOS NA IDENTIFICAÇÃO DE VEÍCULOS UTILIZANDO <i>HAAR</i> <i>FEATURES</i> E <i>HAAR FEATURES</i> + <i>PGH</i>	33
TABELA 3. 5 - RESULTADOS DA IDENTIFICAÇÃO DE VEÍCULOS DO ALGORITMO DE HAN <i>ET AL</i>	33
TABELA 3. 6 - RESULTADOS OBTIDOS POR GOLIGHTLY <i>ET AL</i>	36
TABELA 3. 7 - RESULTADOS OBTIDOS POR LIU NA IDENTIFICAÇÃO DE POSTES ELÉCTRICOS E NINHOS DE CEGONHA .	39
TABELA 3. 8 - RESULTADOS DA DETECÇÃO DE POSTES E NINHOS DO ALGORITMO DE JOÃO FERNANDES.....	41
TABELA 5. 1 – COMPARAÇÃO DOS RESULTADOS OBTIDOS COM OS PROTÓTIPOS EXISTENTES.....	69
TABELA 5. 2 - RESULTADOS DA IDENTIFICAÇÃO DE POSTES ELÉCTRICOS EM VÍDEOS.....	70
TABELA 5. 3 - QUANTIDADE DE NINHOS EXISTENTES NOS POSTES IDENTIFICADOS.....	72
TABELA 5. 4 - INCORRECTA IDENTIFICAÇÃO DE UMA SECÇÃO DO POSTE COMO CONTENDO UM NINHO.....	73
TABELA 5. 5 - COMPARAÇÃO DOS RESULTADOS DOS PROTÓTIPOS EXISTENTES NA DETECÇÃO DE NINHOS.	74
TABELA 5. 6 - RESULTADOS DA DETECÇÃO DE NINHOS DE CEGONHA EM FICHEIROS DE VÍDEO.	75
TABELA 5. 7 - RESULTADOS DOS DIFERENTES TRABALHOS QUE UTILIZAM <i>HAAR FEATURES</i>	75

Introdução

Nesta tese, desenvolvida no âmbito do mestrado em engenharia informática da FCT / Universidade Nova de Lisboa, em colaboração com a empresa Albatroz Engenharia [Albatroz, 2012], pretende-se identificar automaticamente ninhos de cegonhas em postes eléctricos a partir de sequências de vídeo.

Neste capítulo é descrito o problema a resolver, contextualizado com a influência dos ninhos de cegonha na rede de transmissão de electricidade e as medidas tomadas pela REN para reduzir as interrupções causadas por cegonhas. No final do capítulo são abordadas algumas das dificuldades encontradas e delineado o esboço do algoritmo desenvolvido como solução ao problema.

1.1 Contexto

Actualmente encontram-se instalados em Portugal continental cerca de 8000 postes de muito alta tensão (MAT) interligados por mais de 8300km de linhas de transporte de energia [REN, 2012]. Devido à sua alta estatura, os postes eléctricos são uns dos locais de eleição por parte das cegonhas para a construção dos seus ninhos. Dados referentes ao *V Censo Nacional de Cegonha Branca* [Rosa, G., *et al*, 2005] indicam que cerca de 40% da totalidade de ninhos existentes encontram-se em postes, dos quais 65,2% eram postes eléctricos (1824 ninhos). Mais, verificou-se que o número de ninhos de cegonha existentes tinha aumentado consideravelmente quando comparado com o recenseamento efectuado em 1994.

Na figura 1.1 são ilustrados os resultados da contabilização de ninhos obtidos durante os recenseamentos efectuados em Portugal. Verifica-se que entre 1984 e 2004, o número de ninhos aumentou mais de 5 vezes. Este incremento é justificado com alterações nas estratégias migratórias e num acréscimo de resíduos sólidos urbanos em lixeiras e aterros sanitários [Rosa, G., *et al*, 2005].

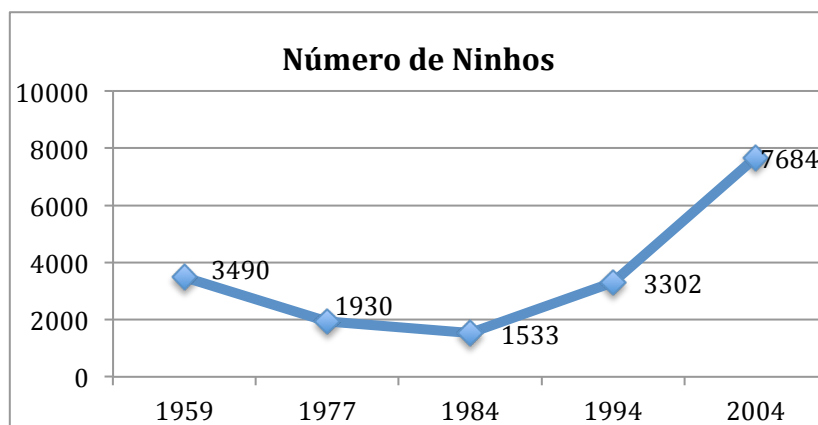


Figura 1. 1 - Evolução do número de ninhos de cegonhas contabilizados nos recenseamentos realizados em Portugal.

A existência de ninhos de cegonha em postes eléctricos poder-se-á revelar problemática. A par de condições atmosféricas adversas (ex. trovoadas), as cegonhas encontram-se entre as principais causas de incidentes na rede nacional de transporte [REN, 2003]. Devido ao aumento da população de cegonhas, a REN tem encetado esforços na monitorização dos seus ninhos. Em concreto, uma das suas maiores preocupações reside em ninhos que se encontram nas proximidades dos isoladores. Estes dispositivos são de grande importância uma vez que *“impedem que a corrente gerada pelos campos eléctricos e magnéticos se desvie do caminho pretendido”* [REN, 2003]. A existência de ninhos nestes locais origina a acumulação de sujidade nos isoladores (dejectos e detritos dos ninhos), o que poderá causar falhas momentâneas na rede de distribuição. Acresce que devido a alterações climáticas, os níveis de pluviosidade têm diminuído nas últimas décadas [Santos, F., *et al*, 2002], o que apenas agrava o problema.

Consciente destas problemáticas, a REN implementou diversas medidas preventivas. Inicialmente foram colocadas bóias e redes de *nylon* nas proximidades dos isoladores, mas a sua eficácia diminuiu com o passar do tempo, à medida que as cegonhas se habituavam à sua existência e os utilizavam em benefício próprio para a construção dos seus ninhos. Posteriormente instalaram-se dispositivos dissuasores de poiso (semelhantes a anemómetros) nestes locais, que apesar de se revelarem relativamente eficazes não resolveram por completo

o problema. Optou-se então pela construção de plataformas dedicadas para ninhos em locais considerados seguros (afastados dos isoladores). Como complemento foram ainda realizadas acções de realocação de ninhos das proximidades dos isoladores para estas mesmas plataformas.

Tabela 1. 1 - Localização dos ninhos em postes MAT.

	Plataformas	Isoladores	Zona de passagem	Outros	Totais
2006	905	111	210	115	1341
2007	1028	142	205	166	1541
2008	1012	70	147	202	1431
2009	1067	64	219	260	1610
2010	1141	66	203	387	1797

Os resultados destas medidas tornam-se evidentes através dos dados presentes na tabela 1.1. Verifica-se que apesar do aumento do número de ninhos em postes eléctricos, a percentagem dos que se encontram nas proximidades dos isoladores tem vindo a diminuir. Em 2006 este valor era de 8,3% tendo recuado para 3,7% em 2010. Esta redução é justificada pela constante instalação de dispositivos preventivos e pelas frequentes acções de realocação de ninhos. Destaque ainda para a quantidade de ninhos existentes em plataformas, com cerca de 63% dos ninhos construídos nestes locais.

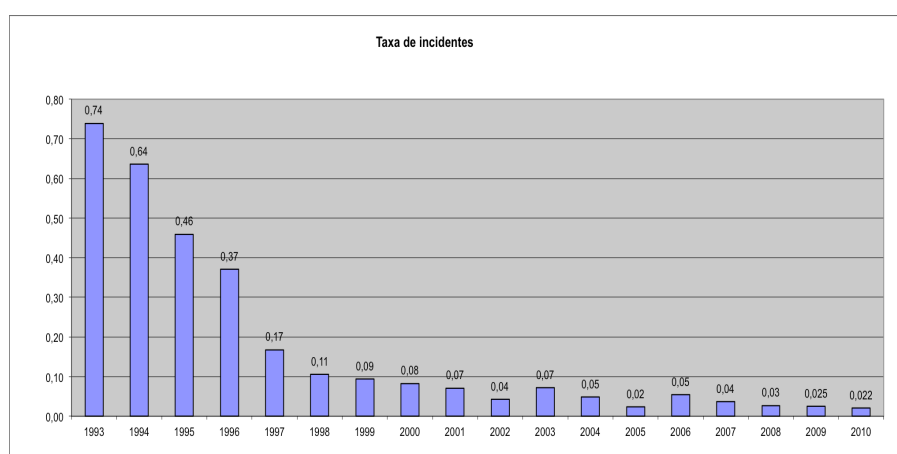


Figura 1. 2 - Evolução da taxa de incidentes causadas por cegonhas nas décadas de 90' e 00'. Apesar do aumento de ninhos, verifica-se uma diminuição dos incidentes causados por cegonhas.

Como resultado das acções levadas a cabo pela REN, a taxa de incidentes causadas por cegonhas tem diminuído. Analisando os dados da figura 1.2, que ilustra a taxa de incidentes em função do número de ninhos existentes e do número de incidentes, verifica-se que durante a década de '90 (altura em que se iniciaram estas medidas preventivas), existiu uma redução significativa nos incidentes causados por cegonhas. Já na década de '00, e apesar do constante incremento do número de cegonhas, a tendência de decréscimo tem-se mantido.

1.2 Descrição do problema

Os dados ilustrados na secção anterior indicam que apesar do sucesso da execução das medidas, continuam a existir alguns incidentes causados por cegonhas. A aposta da REN tem passado pelo aumento de acções de realocação de ninhos.

Actualmente são efectuados dois tipos de inspecções aos postes eléctricos – uma dedicada às linhas de alta tensão e outra para a identificação de ninhos de cegonhas. Tradicionalmente, estas inspecções são efectuadas com recurso a um helicóptero e implicam custos avultados. No caso das inspecções dedicadas à identificação de ninhos de cegonha, um operador captura imagens dos postes, prestando o devido destaque aos ninhos de cegonha, que posteriormente terão que ser analisadas individualmente e reconhecidas as situações em que se deverá proceder a acções de realocação. Esta operação é relativamente demorada e está condicionada temporalmente. De modo a reduzir o impacto que estas acções causam às cegonhas, estas não poderão ocorrer durante a fase de nidificação.

Com os avanços tecnológicos registados nos últimos anos, em particular em técnicas de processamento de imagem, a detecção automática de objectos é hoje uma opção cada vez mais utilizada. A existência de uma ferramenta capaz de identificar ninhos de cegonhas em postes eléctricos poderá simplificar as acções de realocação de ninhos. Entre as diversas vantagens, encontram-se:

- Eliminação da necessidade de vistorias dedicadas à identificação de ninhos, equipando uma câmara no helicóptero utilizado nas vistorias às linha de alta tensão.

- Redução dos custos envolvidos nas acções de monitorização.
- Diminuição dos impactos causados nas cegonhas durante estas acções, uma vez que o ruído causado pelo helicóptero afecta negativamente as mesmas.
- Rápida identificação dos ninhos.
- Possível redução do número de incidentes causados por cegonhas.

Torna-se então evidente a necessidade de uma aplicação capaz de processar uma sequência de vídeo ou imagens e identificar os ninhos de cegonhas existentes nas mesmas. Esta solução traria benefícios não só à REN, mas também aos consumidores.

1.3 Objecto de Estudo

Nesta secção será efectuada uma apresentação sumária das partes envolvidas nesta tese. Inicialmente serão introduzidas algumas noções elementares sobre cegonhas e os seus ninhos. Posteriormente será identificado o modelo de poste eléctrico adoptado nesta tese.

1.3.1 Cegonha Branca

As cegonhas brancas (*Ciconia Ciconia*) são aves de grande porte, pesando em média cerca de 3kg e podendo atingir os 215cm de envergadura e 115cm de comprimento. Para a construção dos seus ninhos, as cegonhas utilizam detritos presentes nas redondezas e a escolha da sua localização é dependente da existência de abundante alimento.

1.3.2 Postes eléctricos

Os postes eléctricos poderão ser agrupados em três categorias: baixa tensão, média tensão e alta/muito alta tensão (MAT). Os postes de baixa e média tensão são utilizados pela empresa Energia de Portugal (EDP) para o forneci-

mento de energia aos seus consumidores, enquanto os de alta/muito alta tensão são utilizados pela REN para o transporte de electricidade pelo país.

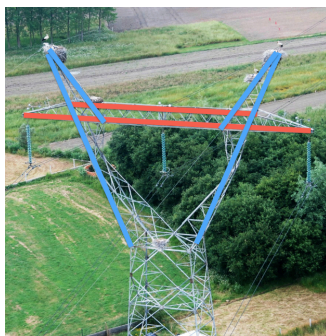


Figura 1.3 - Variante de poste MAT utilizado nesta tese.

No âmbito desta tese, serão apenas considerados os postes MAT, em específico uma das suas variantes mais comuns. Na figura 1.3 é apresentado o modelo de poste empregue, sendo essencialmente caracterizado por duas secções diagonais intersectados por uma horizontal. Este tipo de postes poderá variar em dimensões e a localização dos mesmos é igualmente díspar (desde campos agrícolas até zonas urbanas)

1.4 Proposta de Solução

Como já referido, o processo de detecção poderá ocorrer em sequências de vídeo ou em imagens individuais. No caso de ficheiros de vídeo, estes serão processados como sendo uma sequência de imagens, ou seja, serão extraídas *frames* em intervalos de 10 e para cada uma destas será testada a existência de postes e ninhos.

Com base nestes dados, a abordagem tomada para a resolução do problema consiste em dividir o mesmo em duas partes – reconhecimento do poste eléctrico e identificação dos ninhos de cegonha contidos no mesmo.

1.4.1 Identificação de postes eléctricos

A identificação de um poste eléctrico será alcançada através do reconhecimento de alguns padrões característicos do mesmo. Em concreto pretende-se identificar a região apresentada na figura 1.3, que é caracterizada pelas arestas

diagonais e horizontais e através dos ângulos existentes entre as mesmas. A extracção das arestas será assegurada através de algumas técnicas de processamento de imagem.

Apesar dos padrões que se pretendem reconhecer não serem usuais na natureza, a diversidade de cenários (*background*) poderá dificultar este processo. Há outro tipo de construções humanas que partilham algumas destas características e que poderão ser incorrectamente identificadas como sendo um poste eléctrico. Deste modo terão que ser garantidas algumas condições adicionais que permitam assegurar com um certo nível de confiança a real existência de um poste.

Havendo uma correcta identificação de um poste, a região que o delimita será assinalada, verificando-se posteriormente a existência de ninhos nesse local. Limitar a pesquisa de ninhos à área que delimita o poste, permitir diminuir a probabilidade de identificações incorrectas. No caso de não ser reconhecido qualquer poste, o processo termina e procede-se à análise da imagem seguinte.

1.4.2 Identificação de ninhos de cegonha

Uma vez detectado um poste, procede-se à identificação de ninhos de cegonha. Este processo é assegurado com recurso a um algoritmo capaz de reconhecer características de um ninho. Como pré-condição, será necessário que o algoritmo acesse uma fase de aprendizagem. Durante este período, um conjunto alargado de imagens positivas (ninhos de cegonha) e negativas (paisagem) são examinadas com o intuito de recolher as particularidades que definem o objecto de estudo. Será com base nesta informação que o algoritmo será capaz de detectar a existência e a localização de um ninho numa dada imagem. Entre algumas das vantagens deste algoritmo encontram-se a capacidade de reconhecimento de ninhos de diversas dimensões e em diferentes ângulos de visualização.

Previamente foram identificados alguns dos problemas existentes no reconhecimento de ninhos de cegonha. O facto de as cegonhas utilizarem detritos existentes na paisagem para a construção dos seus ninhos, poderá originar que o algoritmo detecte alguns falsos positivos na paisagem. A restrição da região de pesquisa à área do poste foi uma das medidas tomadas para minorar o problema. Existem ainda outras situações que poderão dificultar o processo de reconhecimento. Entre as ocorrências mais comuns encontram-se os ninhos parci-

almente ocultos (construídos ‘dentro’ da estrutura do poste) e os casos em que dois (ou mais) ninhos se sobrepõem. Neste último cenário, o conjunto de ninhos será identificado como um único.

No final o verdadeiro desafio assentará na capacidade de detectar o maior número possível de ninhos, mantendo um baixo nível de falsos positivos.

1.5 Contribuições

Os benefícios de automaticamente identificar ninhos de cegonha em postes eléctricos, foram mencionados na secção 1.2. Sumariamente, possibilita uma redução de custos em vistorias e, possivelmente, de incidentes causados por cegonhas.

Previamente existiram outros trabalhos que endereçaram o problema em questão. Nesta tese é proposta uma nova abordagem, que se distingue no método de reconhecimento de ninhos. Esta abordagem não enfrenta limitações em termos de localização dos ninhos (em relação ao poste), dimensão ou ângulo de visualização. Apesar de esta nova metodologia introduzir maior complexidade de processamento, permite a obtenção de melhores resultados.

O processo de identificação de postes eléctricos baseia-se em princípios já explorados, mas distingue-se por uma análise exaustiva, onde cada imagem é processada 4 vezes utilizando diferentes técnicas. Desta forma é possível identificar um maior número de postes, mesmo em cenários (*background*) mais complexos.

O principal objectivo desta tese é disponibilizar uma ferramenta o mais eficaz possível na identificação de ninhos de cegonha e postes eléctricos, mesmo que o desempenho seja negativamente afectado.

1.6 Estrutura

O restante documento encontra-se estruturado da seguinte forma:

- Capítulo 2: Aborda conceitos necessários para uma melhor compreensão do algoritmo proposto, descrevendo o funcionamento das técnicas de processamento de imagem empregues nesta tese.
- Capítulo 3: É efectuada uma análise crítica a trabalhos relacionados com o tema desta tese.
- Capítulo 4: Descreve em detalhe o funcionamento do algoritmo desenvolvido.
- Capítulo 5: Apresenta os resultados dos testes efectuados, comparando os mesmos com os outros trabalhos existentes
- Capítulo 6: São retiradas as conclusões ao trabalho efectuado e abordados possíveis melhoramentos a realizar no futuro
- Anexo: Integra um conjunto de imagens utilizadas para os testes efectuados.

Conceitos Fundamentais

Neste capítulo são descritos os algoritmos de processamento de imagem utilizados nesta tese. A percepção do funcionamento destes algoritmos permitirá uma melhor compreensão da solução proposta nesta dissertação.

Inicia-se o capítulo com algumas noções sobre imagens. Na segunda secção são diferenciados filtros lineares de não lineares, apresentando-se alguns dos filtros de imagem mais comuns. Na terceira e quarta secção é explicado em detalhe o funcionamento de duas técnicas utilizadas no algoritmo proposto. E por fim na última secção é apresentada uma biblioteca bastante utilizada no processamento de imagem.

2.1 Imagens de rastreio e imagens vectoriais

Uma imagem digital consiste numa representação a duas dimensões de um cenário, podendo-se agrupar em dois conjuntos: rastreio e vectoriais.

As imagens de rastreio são compostas por *pixéis* sob a forma de uma matriz. Cada *pixel* possui uma cor, representando um pequeno elemento da imagem. A qualidade de uma imagem é medida pela sua resolução, podendo ser referida por *pixels per inch* (PPI) ou *dots per inch* (DPI).

Entre as principais vantagens, destacam-se as seguintes:

- Alto nível de detalhe, especialmente útil para fotografia. Naturalmente a qualidade da imagem estará dependente da quantidade de *pixéis* da mesma.

- Facilidade de edição, nomeadamente através da utilização de filtros.

Apesar de este tipo de imagens ser amplamente utilizado, apresenta também algumas desvantagens:

- O escalonamento (*zoom*) implica uma perda de qualidade.
- Dependendo das dimensões da imagem, existe uma grande quantidade de informação que é necessário armazenar.

De notar que o impacto destas limitações poderá não ser muito significativo com o surgimento de imagens de alta definição e algoritmos de compressão.

No segundo conjunto de imagens, inserem-se as vectoriais que são criadas com recurso a fórmulas matemáticas. No limite, qualquer figura pode ser descrita como um conjunto de pontos interligados por rectas (ou curvas). Este tipo de imagens é bastante utilizado em áreas mais artísticas, como na criação de logótipos.

Comparativamente às imagens de rastreio, este tipo de imagens apresenta diversas vantagens [Harris, A., 2003]:

- Escaláveis: As imagens são facilmente escaláveis sem perda de qualidade
- Pequena dimensão: O tamanho físico da imagem depende da complexidade dos seus componentes e não das tonalidades ou dimensões (largura e altura).
- Manipulação Individual: Uma vez que uma imagem é composta por vários componentes descritos por fórmulas matemáticas é possível manipular individualmente cada um desses componentes.

No entanto estas imagens também apresentam algumas desvantagens:

- Reprodução de curvas: As curvas complexas são de difícil reprodução.
- 'Lentidão': O processo de *loading* é mais demorado do que nas imagens de rastreio.
- Aspecto *cartoon*: Não é possível apresentar as mesmas tonalidades de cor presentes em imagens de rastreio, perdendo algum do seu realismo.

É essencialmente devido a estas limitações que as imagens de rastreio são mais populares.

2.2 Filtros de Imagem

No processamento de imagem, os filtros são utilizados de modo a transformar uma imagem X em X' . De entre as operações possíveis, poder-se-á remover ruído [Buades, A., *et al* 2005] ou realçar determinadas propriedades (ex. arestas). Consideremos dois tipos de filtros [Glasbey, C., Horgan, G., 1995]: Lineares e não lineares.

2.2.1 Filtros Lineares

Um filtro diz-se linear se respeitar as seguintes condições:

- **Linearidade:** O resultado da soma de duas imagens é igual à sua soma individual.

$$f(I1+I2) = f(I1) + f(I2)$$

- **Shift-Invariance:** O resultado de aplicar o filtro não altera o padrão formado na imagem.

A utilização deste tipo de filtros é especialmente útil para a remoção de ruído gaussiano [Buades, A., *et al* 2005]. Este procedimento é denominado por *smoothing* e existem diversos filtros passíveis de escolha:

- **Averaging Filter** (ou *Mean filter*): Uma das características das imagens reside na proximidade entre o valor de um *pixel* com o dos seus vizinhos. Recalcular o valor de um *pixel* com base na média dos seus vizinhos poderá ser uma forma de eliminar alguns tipos de ruído, como *Salt and Pepper* [Buades, A., *et al* 2005].
- **Gaussian Smoothing Filter:** Este é um dos filtros mais utilizados na remoção de ruído, principalmente em algoritmos de detecção de arestas. O filtro consiste em atribuir pesos diferentes aos pixéis de uma imagem consoante uma função gaussiana.

Os filtros aqui apresentados são apenas uma pequena amostra dos vários filtros existentes. Realça-se o filtro gaussiano, utilizado pelo *Canny edge detector*.

2.2.2 Filtros não lineares

Intuitivamente, um filtro não linear é aquele cujo resultado não é obtido através de uma função linear. De entre os filtros não lineares destaca-se o *median filter*. Funciona de forma semelhante aos filtros lineares no sentido que o filtro (janela) é sucessivamente aplicado aos diferentes pixéis da imagem. No entanto não são atribuídos pesos à vizinhança. Em vez disso os valores da janela são ordenados de forma crescente, sendo que o valor escolhido é o do meio. Este tipo de filtros apresenta algumas vantagens em relação aos filtros lineares: Permitem manter um maior nível de detalhe e não são negativamente afectados por variações extremas de pixéis vizinhos. No entanto poderão ser menos eficientes na remoção de ruído e acarretam custos computacionais mais elevados.

2.3 Canny Edge Detector

Apresentado em 1986 por John Canny, o *Canny edge detector* [Canny, J., 1986] é ainda nos dias de hoje um dos algoritmos mais utilizados na reconhecimento de arestas. De forma resumida, este algoritmo identifica os locais de uma imagem onde existe variação abrupta da intensidade dos pixéis. Para a elaboração do seu algoritmo, John Canny estabeleceu três objectivos:

- Taxa de erro reduzida.
- Correcta localização das arestas.
- Inexistência de arestas duplicadas.

O cumprimento destes objectivos permitiu que o *Canny edge detector* se superiorizasse a outros algoritmos de extracção de arestas. No essencial, o funcionamento deste algoritmo poderá ser decomposto em 4 etapas:

- Redução de ruído. O ruído poderá afectar negativamente o desempenho do algoritmo, pelo que é utilizado um filtro (tipicamente gaussiano) com o intuito de reduzir o seu impacto. Na figura 2.1 é ilustrada a aplicação de um filtro gaussiano de matriz 11x11. Verifica-se que a imagem resultante se encontra 'turva' (*blurred*) quando comparada com a original, o que poderá conduzir a melhores resultados. Deve ser tido em considera-

ção que a quantidade de arestas extraídas pelo algoritmo será inferior, podendo-se perder alguma informação importante. Deste modo, a escolha da matriz utilizada deverá ter este factor em consideração. Uma matriz de menores dimensões (ex. 5x5) permitirá a obtenção de um maior número de arestas, mas será menos eficiente na redução de ruído.



Figura 2. 1 - Utilização de um filtro gaussiano de matriz 11x11. a) Imagem original. b) Imagem resultante.

- Identificação de regiões de interesse através do cálculo do ângulo e magnitude do gradiente. Para cada ponto será calculada a derivada em x e em y . Com base nestes valores será possível obter a magnitude e ângulo do gradiente:

$$G = \sqrt{D_x^2(x,y) + D_y^2(x,y)}$$

Cálculo do gradiente

$$\theta = \arctan \frac{D_y(x,y)}{D_x(x,y)}$$

Cálculo do ângulo do gradiente

Sabendo o ângulo do gradiente este deverá ser arredondo através de uma das seguintes condições:

- $\theta(x,y) \in [0^\circ, 22.5^\circ[\vee [157.5^\circ, 202.5^\circ[\vee [337.5^\circ, 0^\circ[\rightarrow \theta = 0^\circ$
- $\theta(x,y) \in [22.5^\circ, 67.5^\circ[\vee [202.5^\circ, 247.5^\circ[\rightarrow \theta = 45^\circ$
- $\theta(x,y) \in [67.5^\circ, 112.5^\circ[\vee [247.5^\circ, 292.5^\circ[\rightarrow \theta = 90^\circ$
- $\theta(x,y) \in [112.5^\circ, 157.5^\circ[\vee [292.5^\circ, 337.5^\circ[\rightarrow \theta = 135^\circ$

- Eliminação de arestas candidatas através de comparação do gradiente com pixels vizinhos (*Non-Maximum Suppression*). Do passo anterior são

extraídas diversas arestas. No entanto algumas são demasiado densas, devendo idealmente possuir uma espessura de apenas um pixel. O objetivo desta etapa passará por escolher os pixéis com maior gradiente, considerando os seus vizinhos:

- Se $\theta(x,y) = 0^\circ$, são examinados os pixéis $(x+1, y)$, (x,y) e $(x-1, y)$.
- Se $\theta(x,y) = 45^\circ$, são examinados os pixéis $(x+1, y+1)$, (x,y) e $(x-1, y-1)$
- Se $\theta(x,y) = 90^\circ$, são examinados os pixéis $(x, y+1)$, (x,y) e $(x, y-1)$
- Se $\theta(x,y) = 135^\circ$, são examinados os pixéis $(x+1, y-1)$, (x,y) e $(x-1, y+1)$

Para cada *pixel* (x,y) , se o gradiente for maior que os restantes dois pixéis vizinhos analisados, este será guardado como sendo uma aresta.

- Eliminação de ruído nas arestas (*Hysteresis Thresholding*). Das arestas identificadas nos passos anteriores, algumas não deverão ser consideradas uma vez que serão ruído. Nesta última etapa estes pixéis deverão ser descartados, eliminando os que possuírem um gradiente inferior a um determinado *threshold*.

No entanto esta hipótese originaria um novo problema. Os pixéis que possuísem um gradiente próximo do *threshold* (imediatamente inferior) seriam eliminados, levando a que na imagem final existissem arestas descontínuas. A solução encontrada consiste em utilizar dois *thresholds* (T_{min} , T_{max}) em vez de apenas um. Cada pixel com gradiente superior a T_{max} é considerado como pertencente a uma aresta. Posteriormente todos os seus vizinhos são analisados de forma recursiva, sendo guardados os que possuírem um gradiente superior a T_{min} .

A eficiência deste algoritmo está intrinsecamente ligada à escolha dos valores T_{min} e T_{max} . A análise da figura 2.2 permite constatar a diferença das arestas extraídas com diferentes valores de *threshold*. Comparando as figuras a) e b) verifica-se uma redução das arestas identificadas na última. Isto ficou-se a dever ao incremento de T_{max} de 700 na figura a) para 900 na figura b), tornando a primeira condição de aceitação da etapa de *Hysteresis Thresholding* mais restritiva. Por outro lado as arestas que foram identificadas em b) possuem o mesmo comprimento que em a), uma vez que se manteve o valor de T_{min} .

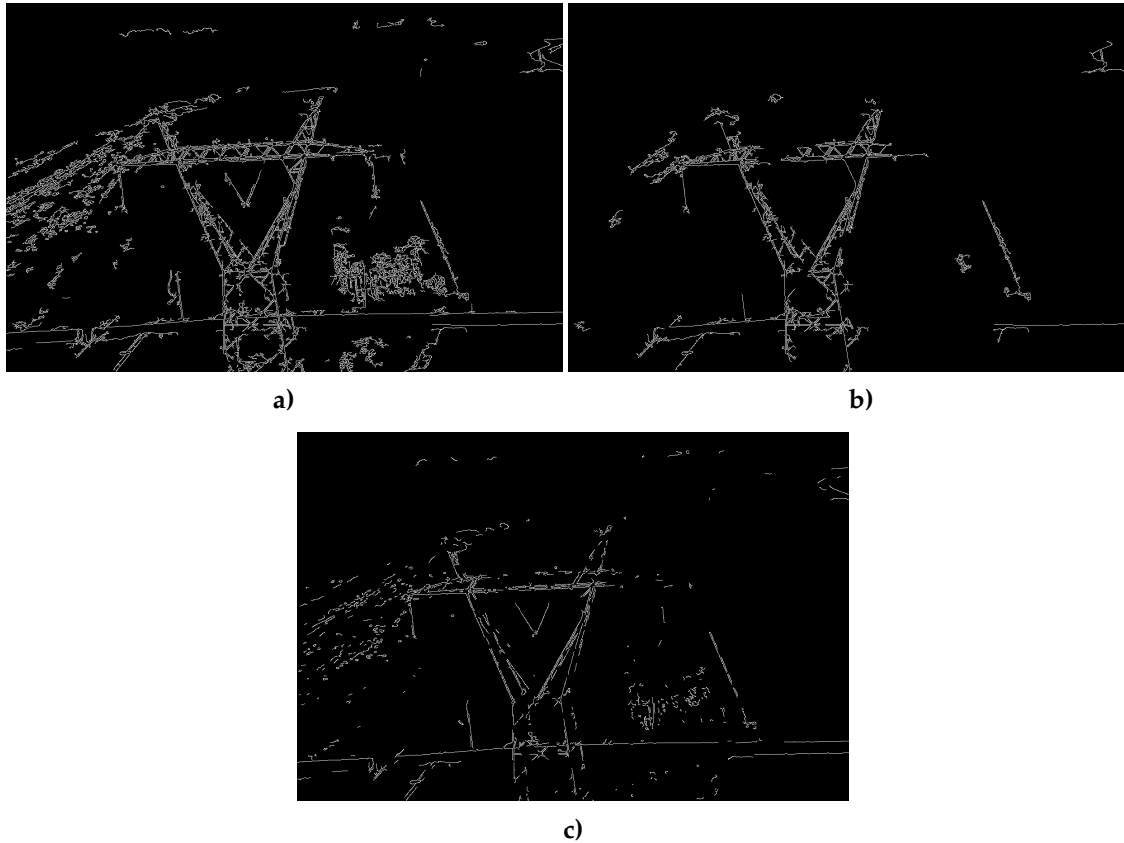


Figura 2. 2 - Resultado das arestas extraídas pelo algoritmo *Canny* aplicando diferentes *thresholds*. a) $T_{min}=200$ e $T_{max}=700$. b) $T_{min}=200$ e $T_{max}=900$. c) $T_{min}=400$ e $T_{max}=700$.

A comparação das figuras a) e c) permite verificar que apesar de terem sido identificadas as mesmas arestas, na figura c) estas possuem um menor comprimento. Esta diferença fica-se a dever ao aumento de T_{min} para 400 na figura c), levando a que alguns dos pixéis vizinhos dos pixéis identificados com gradiente superior a T_{max} , não tenham sido considerados.

2.4 Hough Transform

A transformada de *Hough* [Duda, R., Hart, P., 1972] é um algoritmo bastante utilizado na extracção de características (*features*) numa imagem. Originalmente desenvolvido por Paul Hough e posteriormente adaptado por Richard Duda e Peter Hart apresenta como pré-condição que as características pretendidas possam ser definidas de forma paramétrica. Tal requisito significa que o algoritmo é especialmente útil na detecção de arestas, ou figuras como círculos ou elipses [Soltany, M., *et al*, 2011].

Considerando um ponto (x, y) de uma imagem, existe um conjunto de rectas que intersectam esse ponto definidas por:

$$y = mx + b \quad (1)$$

podendo estas mesmas rectas ser representadas por:

$$b = -mx + y \quad (2)$$

Observando as expressões anteriores, verifica-se que cada recta que intersecta o ponto (x, y) da representação (1), corresponde a um ponto em (2). Esta última expressão é denominada por espaço de *Hough* (*Hough Space*).

Observando a figura 2.3, o conceito anterior poderá ser estendido para dois pontos P e Q . De entre o conjunto de rectas que intersectam cada um destes pontos, existe apenas uma que engloba simultaneamente os dois pontos. Esta recta é representada no espaço de *Hough*, no ponto de intersecção (b, m) das rectas p e q .

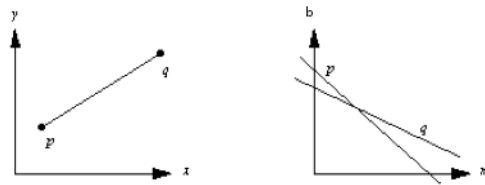


Figura 2.3 - Identificação da recta que intersecta dois pontos no espaço de *Hough*.

A anterior representação poderá originar elevados custos computacionais para as rectas verticais, uma vez que existe um conjunto infinito de valores de b e m . De modo a contornar este problema, as rectas deverão ser definidas em coordenadas polares, segundo:

$$r = x * \cos(\theta) + y * \sin(\theta) \quad (3)$$

onde r mede a distância entre a recta e a origem e θ o respectivo ângulo. A utilização de coordenadas polares implica uma alteração da representação do espaço de *Hough*. Como ilustrado na figura 2.4, as anteriores rectas são agora substituídas por curvas. No entanto, o funcionamento do algoritmo mantém-se:

Considerando três pontos P , Q e R , o ponto de intersecção (r, θ) das três curvas no espaço de *Hough*, definirá a recta que intersecte esses mesmos pontos.

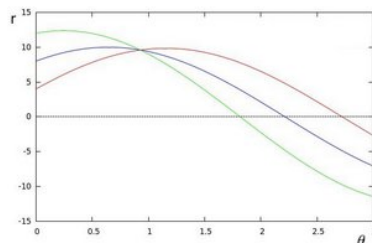


Figura 2. 4 - Representação do espaço de *Hough* em coordenadas polares.

Como pré requisito ao algoritmo, será necessário determinar os pontos de interesse na imagem, podendo ser utilizado um algoritmo de detecção de arestas, como o *Canny edge detector*. Posteriormente é utilizado um acumulador com recurso a uma matriz (r, θ) , decorrendo um processo de votação no espaço de *Hough*. Para cada ponto de interesse é incrementada a respectiva entrada no acumulador e terminada esta análise são escolhidas as entradas de interesse com base num *threshold* previamente definido. O resultado deste processo irá determinar as arestas existentes na imagem.

2.5 Haar like features

Em 2001, Paul Viola e Michael Jones [Viola, P., Jones, M., 2001] adoptaram a ideia de utilizar *Haar Wavelets* propondo um novo conceito denominado por *Haar-like features*. Desde então a sua teoria tem sido amplamente utilizada na detecção de objectos, com especial relevo no reconhecimento facial. Na sua versão inicial, a proposta de Viola e Jones distinguia-se em três componentes:

- Introdução de uma nova representação de imagem (*Integral Images* [Derpanis, K., 2007]) que permitia uma rápida recolha de características presentes numa imagem.
- Adopção de um algoritmo de aprendizagem (baseado no *AdaBoost* [Freund, Y., et al. 1995]) que possibilitava a construção de um classificador bastante eficiente a partir de um pequeno conjunto de características chave.
- Combinação de classificadores mais complexos em “cascata” (*cascade*),

que permitia ignorar as secções de fundo (*background*) e concentrar os esforços nas regiões contendo objectos de interesse.

Uma das dificuldades na detecção de objectos advém do contexto em que os mesmos se inserem. Estes poderão encontrar-se dispostos em diferentes ângulos ou contidos em cenários com diversos níveis de luminosidade. Com o intuito de obter bons resultados de detecção, Viola e Jones propuseram a utilização de *haar-like features* que seriam capazes de lidar com estas variâncias. Estas *features* baseavam-se nas mesmas usadas por Papageorgiou [Papageorgiou, C., et al., 1998], destacando-se as *two-rectangle feature* e as *three-rectangle feature*. Estas *features* distinguem-se pelo tipo de características que se pretende detectar, existindo conjuntos específicos para arestas (*two-rectangle features*) e para rectas (*three-rectangle features*).

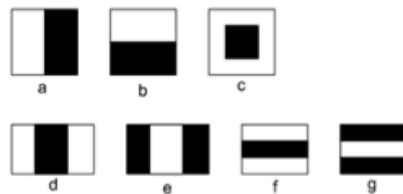


Figura 2. 5 – Conjunto de algumas *features* adoptadas por Viola e Jones. a) e b) - *edge features*. d), e), f) e g) - *line features*.

Na figura 2.5 são apresentadas algumas *features* utilizadas por Viola e Jones. Cada um destes rectângulos é aplicado a uma imagem, sendo obtido o seu valor com base nos pixéis sobrepostos na imagem. O cálculo, para cada um dos conjuntos, será:

- *Two-rectangle feature*: Diferença da soma dos pixéis da imagem na região a preto com a região a branco.
- *Three-rectangle feature*: Subtracção da soma dos pixéis da região central á soma dos pixéis dos rectângulos exteriores.

Apesar dos benefícios que a utilização de este tipo de *features* poderá apresentar, os custos de computação tendem a crescer exponencialmente para imagens de grande dimensão (ex. Imagens de alta definição). Considerando uma imagem de 400x400, existirão 142129 subjanelas de dimensão 24x24 $((400-24+1)*(400-24+1))$. Acresce que para cada uma destas subjanelas existem cerca

de 180 000 *features*, o que corresponde a um total de 25.583.220.000 *features* que terão de ser calculadas.

As optimizações propostas por Viola e Jones incidem na redução de *features* necessárias, adaptando o *Adaboost*, e nos esforços do cálculo de cada *feature* utilizando *integral images*.

2.5.1 Integral Images

Em 1984, Frank Crow apresentou um novo método que permitia um rápido processamento de imagens, utilizando uma representação intermédia - *summed area tables* [Crow, F., 1984]. Esta nova representação, também conhecida como *integral image*, ganhou popularidade após a sua adopção por parte de Viola e Jones.

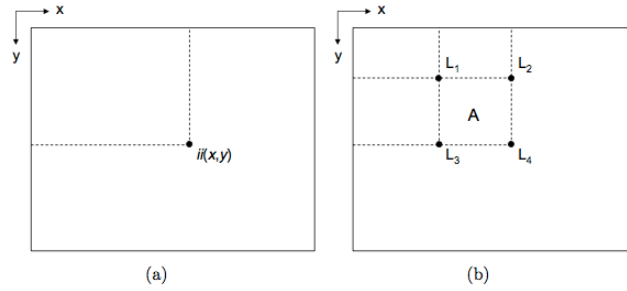


Figura 2. 6 - Representação de uma *integral image*. a) Restrição a um ponto. b) Restrição de uma área.

Consideremos a figura 2.6 – a), em particular o ponto $ii(x, y)$. Numa *integral image*, o valor deste ponto corresponderá à soma de todos os pixels superiores à esquerda. Formalmente, o valor de cada ponto será obtido segundo:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

onde $i(x', y')$ corresponde a um pixel na imagem original. Já o cálculo da *integral image* completa é alcançado através das seguintes recorrências:

$$\begin{aligned} s(x, y) &= s(x, y-1) + i(x, y), \\ ii(x, y) &= ii(x-1, y) + s(x, y), \\ s(x, -1) &= 0, \\ ii(-1, y) &= 0 \end{aligned}$$

onde $s(x,y)$ representa a soma das linhas (*rows*). Desta forma as *integral images* permitem o rápido cálculo de uma qualquer região rectangular, sendo necessárias apenas quatro referências. Na figura 2.6 – b), o cálculo da região A é obtido através de:

$$A = L4 + L1 - (L2 + L3)$$

Nos moldes apresentados, as *integral images* apenas são úteis para o cálculo de áreas de figuras cujas arestas são paralelas ao eixo de coordenadas. Em 2002 Lienhart e Maydt estenderam o conceito para rectângulos orientados em 45° a que chamaram de *rotated summed area table* [Lienhart, R., Maydt, J., 2002]. Apesar da natural vantagem desta nova representação, para o cálculo da *integral image* são necessárias duas análises da imagem original.

2.5.2 AdaBoost

Desenvolvido por Yoav Freund e Robert Schapire em 1996, o *AdaBoost* (*Adaptive Boosting*) é bastante utilizado em algoritmos de aprendizagem de modo a melhorar a performance.

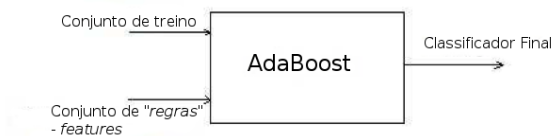


Figura 2. 7 - Funcionamento do AdaBoost.

Como ilustrado na figura 2.7, a ideia consiste em combinar um conjunto de regras (ou fracos classificadores - *weak classifiers*) e dados de treino (amostras positivas e negativas) de modo a construir um modelo (ou classificador forte - *strong classifier*) capaz de identificar um determinado objecto. No caso do algoritmo proposto por Viola e Jones, o conjunto de regras é construído com base nas *haar-like features*, devendo as mesmas ser capazes de distinguir as amostras positivas das negativas.

Como abordado em [Meynet, J., 2003] o algoritmo apresenta algumas vantagens:

- Não é necessário qualquer conhecimento prévio da estrutura ou caracterís-

tica do objecto que se pretende detectar. Apenas é exigido um conjunto de treino (imagens) e outro de “regras” (funções de classificação).

- O algoritmo funciona de forma adaptativa. Em cada etapa do processo de aprendizagem, o conjunto de treino (positivo e negativo) é testado. Existindo amostras classificadas erroneamente, são atribuídos pesos superiores de modo a incrementar a capacidade de diferenciação nas etapas seguintes.
- O erro do processo de aprendizagem tende para zero (teoricamente).

No entanto são também identificadas algumas desvantagens:

- A qualidade dos resultados depende do conjunto de dados e regras.
- O processo de treino é demorado, uma vez que a cada etapa o algoritmo testa todas as regras em todas as amostras.

Verifica-se que apesar das suas limitações o algoritmo deverá ser capaz de identificar o objecto pretendido com um bom grau de exactidão.

Uma vez apresentada uma visão geral do algoritmo, é importante compreender o funcionamento do mesmo, que consiste nas seguintes fases:

1. É dado um conjunto de treino: $(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)$ em que x_n se refere à amostra n , e y_n representa a sua classe (1 para as amostras positivas e 0 para as negativas).
2. Inicializam-se os pesos a atribuir às amostras. Para as positivas é atribuído um peso de $w = \frac{1}{2l}$ em que l representa o número de amostras positivas. Para as amostras negativas é atribuído um peso $w = \frac{1}{2m}$ onde m representa o número de amostras negativas.
3. Para cada etapa $t1$ até T :
 - a) Normaliza-se os pesos das amostras:
$$W_{t,i} \leftarrow \frac{W_{t,i}}{\sum_{j=1}^n w_{t,j}}$$
 - b) Para cada regra i , constrói-se um classificador h_j e calcula-se o seu erro estimado ε_t .

$$\varepsilon_j = \sum_i^n w_{t,i} |h_j(x_i) - y_i|$$

- c) De entre os classificadores, escolhe-se o que possuir a menor taxa de erro e actualizam-se os pesos.

$$w_{t+1,i} = w_{t,i} \beta_t^{1-\varepsilon_t}, \quad \beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$$

em que $\varepsilon_t = 0$, se a amostra é classificada correctamente. Caso contrário será igual a 1.

Terminadas todas as etapas, o classificador final será:

$$f(x) = \begin{cases} 1, & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha \\ 0 & \end{cases}$$

em que $\alpha = \log \left(\frac{1}{\beta_t} \right)$.

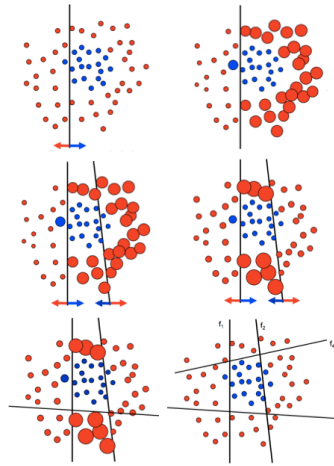


Figura 2. 8 - Processo de treino do classificador.

É possível observar na figura 2.8 as diferentes fases do processo de treino, onde as figuras azuis correspondem a amostras positivas e a vermelho as negativas. Para cada etapa são actualizados os pesos das amostras, sendo atribuído um peso superior às incorrectamente classificadas. Este passo é importante uma vez que permitirá uma melhor diferenciação das amostras nas etapas seguintes. Neste exemplo, o processo de treino era constituído por quatro etapas, sendo

possível verificar que existiu uma boa distinção das amostras positivas das negativas. Como já referido, aumentando o número de etapas a taxa de erros tenderia para zero.

2.5.3 Cascade Classifier

Uma das características do algoritmo proposto por Viola e Jones consistia em combinar classificadores em “cascata” (*cascade*). O objectivo seria de aumentar a taxa de detecção, aliada a uma redução no tempo de computação.

A ideia seria que as primeiras etapas fossem menos complexas e capazes de rejeitar bastantes amostras negativas, ao mesmo tempo que validavam praticamente todas as amostras positivas. Estas últimas seriam utilizadas nas etapas seguintes, reduzindo-se a taxa de erro.

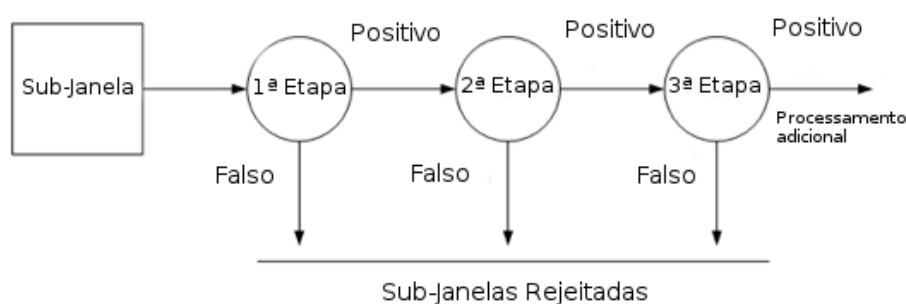


Figura 2. 9 - Funcionamento do classificador em 'cascata'.

A figura 2.9 ilustra o funcionamento do classificador em “cascata”. Uma vez que a maior parte das janelas são rejeitadas nas etapas iniciais, o número de janelas a analisar nas etapas seguintes é significativamente menor. No entanto o processamento das janelas remanescentes torna-se gradualmente mais complexo, pelo que mesmo existindo um número de janelas significativamente superior nas etapas iniciais, as etapas mais avançadas acarretam um tempo de processamento superior.

2.6 OpenCV

O *OpenCV* (*Open Source Compute Vision*) é uma biblioteca que disponibiliza ferramentas à visão computacional (*computer vision*). Desenvolvida pela *Intel* em 1999, após diversas versões *beta*, recebeu a sua primeira versão oficial

(1.0) em 1999. Actualmente encontra-se na versão 2.4.5, suportando diversas plataformas. Entre as diversas funcionalidades destacam-se algoritmos de:

- Processamento de imagem
- Análise vídeo
- Detecção de objectos
- Reconhecimento facial
- Aprendizagem

Actualmente esta biblioteca é amplamente utilizada nas mais variadas temáticas. Em [Jia, X., 2010] diversos algoritmos do *OpenCV* foram aplicados na detecção de defeitos fabris. Em [Fan, X., et al, 2012] foi desenvolvido um sistema de reconhecimento facial adaptando a implementação do *AdaBoost* para construir um classificador com base em *haar features*. Já em [Wafi, Z., et al, 2010] são empregues, entre outros, o *Canny edge detector* e a suavização gaussiana num sistema de videovigilância de auto-estradas. Todos estes algoritmos encontram-se disponíveis no *OpenCV*.

Trabalho Relacionado

No capítulo anterior foram introduzidas diversas técnicas que permitirão uma melhor compreensão dos algoritmos desenvolvidos nesta tese. Neste capítulo são analisados diversos trabalhos que empregam essas mesmas técnicas, encontrando-se agrupados em duas secções: extracção de arestas e detecção de objectos. A última secção é referente às soluções existentes para o problema endereçado nesta dissertação.

Na primeira secção são abordados trabalhos que utilizam os algoritmos *Canny edge detector* e transformada de *Hough*, como meio de extracção de arestas em imagens. É efectuado um enquadramento do problema endereçado e da solução proposta, finalizando-se com uma discussão dos resultados obtidos.

A segunda secção é dedicada à temática de detecção de objectos, em particular a trabalhos que empregam as *haar like features*. São também efectuadas considerações sobre outros algoritmos de detecção de objectos, que poderiam ser utilizados no contexto desta tese.

Por fim o capítulo é finalizado com a apresentação dos dois protótipos existentes. São analisadas as metodologias destas soluções, assim como os resultados obtidos.

3.1 Identificação e Extracção de arestas

Para o reconhecimento de objectos em imagens, uma possível solução consiste em identificar as arestas contidas nas mesmas. Esta metodologia é especialmente eficaz para objectos que possam ser expressos por fórmulas mate-

máticas simples, como rectângulos ou círculos. Durante os últimos anos têm sido apresentados diversos algoritmos capazes de atingir este propósito, sendo o *Canny edge detector* um dos mais utilizados.

Uma área onde este tipo de técnica tem sido utilizada é na segurança rodoviária, quer seja na identificação de automóveis [Fields, A., Green, R., 2012], sinais de trânsito [Cai, L., *et al*, 2010] ou faixas de rodagem [Daigavane, P., Bajaj, P., 2010]. Em [Fields, A., Green, R., 2012] é apresentado um algoritmo capaz de reconhecer veículos em ficheiros vídeo, constituído pelas seguintes fases:

- Para cada *frame* do vídeo:
 - Converter a imagem de RGB para *gray scale*.
 - Identificar arestas utilizando o *Canny edge detector*.
 - Restringir a imagem à região contendo a estrada.
 - Utilizar a transformada de *Hough* para identificar zonas com uma grande concentração de arestas horizontais (potenciais automóveis).
 - Filtrar as zonas previamente identificadas com base na quantidade de arestas horizontais e na distância entre as mesmas.
 - Caso o número de arestas seja insuficiente, identificar a *frame* como potencialmente contendo um automóvel.

Por fim apenas serão considerados os veículos identificados mais próximos, uma vez que o principal objectivo do algoritmo é servir como um sistema de assistência ao condutor, monitorizando a variação da distância para o veículo que o precede. No entanto existem algumas limitações de utilização, sendo a principal, a assunção que a estrada será plana e em linha recta.

Tabela 3. 1 - Resultados da identificação de veículos [Fields, A., Green, R., 2012].

	Não existe veículo	Possível veículo	Existe veículo
Correcto	1228	488	197
Falso	420	18	0
Taxa de Sucesso	75%	96%	100%

Para os testes foi utilizado um ficheiro de vídeo capturado por um veículo da polícia, contendo sequências em alta e baixa velocidade. Os resultados obti-

dos são apresentados na tabela 3.1, onde cada *frame* foi classificada em uma de três classes possíveis: ‘Não existe veículo’, ‘Possível existência de veículo’ e ‘Existe veículo’. Verifica-se que em situações de estrada plana, em linha recta e com cenários (*background*) pouco complexos, o algoritmo obtém uma taxa de sucesso de 100% na detecção de veículos.

No entanto foram enfrentadas algumas dificuldades com os falsos negativos – casos em que existe um veículo e o mesmo não é identificado. Isto ocorreu, essencialmente, em situações em que o veículo se encontrava a uma grande distância da câmara ou perto das extremidades laterais da estrada, em situações de curva ou em cruzamentos. No final, combinando os diferentes cenários possíveis (estradas rectas, com curvas e cruzamentos), o algoritmo obteve uma taxa de detecção de sensivelmente 80%.

Ainda na mesma temática de identificação de veículos, existem outras abordagens possíveis. Em [Mousa, A., 2012] é aproveitada uma característica presente em todos os veículos – matrícula. A identificação deste elemento é efectuada com recurso ao *Canny edge detector* e assenta no reconhecimento do seu padrão: “A suitable description of the license plate is based on solidity, the ratio between the height and width, convex area, and bounding box.”

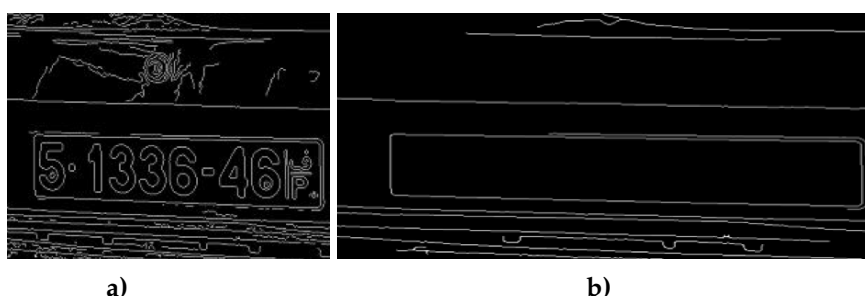


Figura 3. 1 - Identificação da matrícula de um automóvel a) Arestas identificadas com o *Canny edge detector*. b) Imagem resultante do processo de filtragem. [Mousa, A., 2012].

Como ilustrado na figura 3.1, a matrícula do veículo é facilmente reconhecível depois de identificar as arestas. A fase seguinte do algoritmo consiste em filtrar as arestas de pequena dimensão, o que facilitará o reconhecimento do padrão pretendido.

O método apresentado deverá ser capaz de identificar matrículas de veículos. No entanto, considerando o padrão que se pretende identificar e descurando os números/letras, poderá apresentar uma taxa de identificações incorrectas elevada.

Para além da detecção de veículos, a possibilidade de detecção da faixa de rodagem tem sido explorada em diversos trabalhos. Em [Daigavane, P., Ba-

jaj, P., 2010] este objectivo é atingido através da utilização do *Canny edge detector* e da transformada de *Hough*. A diferença do processo de reconhecimento para o apresentado em [Fields, A., Green, R., 2012] reside na utilização de um algoritmo de optimização – *Ant Colony Optimization* [Lu, D., Chen, C., 2008] – após a extracção das arestas. No entanto, e tal como em [Fields, A., Green, R., 2012], o algoritmo apenas é eficaz em estradas rectas.

As aplicações do *Canny edge detector* e da transformada de *Hough* não se cingem à segurança rodoviária. Na realidade estes algoritmos são utilizados para os mais variados fins. Em [Qu, Y., et al., 2005] são utilizados para a identificação de pistas de aeroportos a partir de imagens aéreas. O algoritmo assenta em duas fases: extracção de arestas e utilização de um classificador SVM [Gunn, S., 1998]. Considera-se a existência de duas pistas se forem extraídas duas arestas paralelas e se as mesmas forem validadas pelo classificador.

Outra área em que estas técnicas podem ser utilizadas é em sistemas de controlo de qualidade. Em [Hocenski, Z., et al, 2006] é apresentado um algoritmo capaz de automaticamente identificar defeitos em azulejos. A utilização do *Canny edge detector* permitiu a correcta identificação de 98% das imperfeições presentes nas 20 imagens utilizadas para testes.

Os trabalhos aqui apresentados são apenas uma pequena amostra da investigação realizada na utilização de algoritmos de extracção de arestas para a detecção de objectos. A grande dificuldade reside na redução de ruído presente nas imagens, mas contornado este problema, os resultados demonstram que poderá ser uma boa solução.

3.2 Detecção de objectos utilizando *Haar-features*

Em 2001 Viola e Jones apresentaram um novo método de reconhecimento de objectos [Viola, P., Jones, M., 2001]. As vantagens e funcionamento geral do algoritmo foram previamente enunciadas no segundo capítulo, sendo que o objectivo seria o reconhecimento eficiente de faces. Este procedimento seria obtido através da utilização de um classificador que havia sido previamente treinado nas seguintes condições:

- 9832 imagens positivas (contendo faces) de dimensão 24x24.
- 10 000 imagens negativas de dimensão 24x24.
- 38 etapas utilizando um total de 6061 *features*.

Através destas premissas, o classificador deveria ser capaz de identificar com algum grau de exactidão as faces presentes numa imagem. Para os testes foi utilizado um conjunto de 130 imagens, contendo um total de 507 faces.

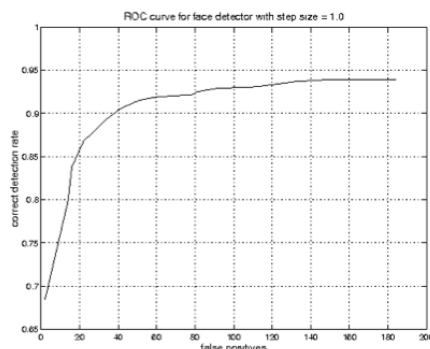


Figura 3. 2 - Resultados obtidos por Viola e Jones no reconhecimento facial [Viola, P., Jones, M., 2001].

Os resultados obtidos encontram-se representados na figura 3.2. Verifica-se que alterando o *threshold* do algoritmo é possível obter taxas de detecção/falsos positivos diferentes. Por exemplo, numa configuração que permite a reconhecimento de 90% das faces existentes, existem apenas 40 detecções incorrectas, o que corresponde a uma taxa de identificação incorrecta de apenas 8%. Para além dos bons resultados, outra vantagem deste algoritmo residia na sua performance, uma vez que permitia uma identificação 15 vezes mais rápida que qualquer abordagem anterior.

Desde a sua apresentação, têm existido diversos trabalhos que estenderam a investigação realizada por Viola e Jones. Em [Ke, L., Kang, J., 2010] pretendia-se não só detectar faces, mas também a região ocular. O algoritmo funcionaria em duas etapas: primeiro seria identificada a região facial e posteriormente a região ocular. Para cada uma das situações foi treinado um classificador utilizando *haar features*, que posteriormente seria testado em dois grupos de imagens: totalmente frontais e com ângulo de deflexão.

Tabela 3. 2 - Resultados da identificação facial do algoritmo de Ke e Kang [Ke, L., Kang, J., 2010].

	Nº de Imagens	Nº de identificações correctas	Taxa de detecção
Grupo 1	264	264	100%
Grupo 2	1477	1436	97.2%

Os resultados do reconhecimento facial podem ser consultados na tabela 3.2. Comparando com os obtidos por Viola e Jones, verifica-se uma melhoria na taxa de detecção. No entanto não foram disponibilizados dados sobre a taxa de identificações incorrectas, pelo que a comparação dos dois algoritmos deverá ser efectuada com algum cuidado.

De notar a diferença entre a taxa de detecção obtida nos dois grupos de imagens utilizados nos testes. Apesar de não ser significativo, o algoritmo enfrentou algumas dificuldades nas imagens do segundo grupo.

Tabela 3. 3 - Resultados da identificação da região ocular do algoritmo de Ke e Kang [Ke, L., Kang, J., 2010].

	Nº de Imagens	Nº de identificações correctas	Taxa de detecção
Grupo 1	264	248	94%
Grupo 2	1477	1190	80.6%

Uma vez identificada a região facial, procede-se à detecção da região ocular, onde os resultados são apresentados na tabela 3.3. Comparativamente com os resultados obtidos no reconhecimento facial, existe uma redução na taxa de detecção, sendo este fenómeno mais significativo nas imagens do segundo grupo. Esta diferença entre os dois grupos de imagens poderá estar relacionada com o conjunto de imagens utilizadas para o treino do classificador.

Apesar de não serem apresentados dados sobre a quantidade de falsos positivos e tendo em conta as taxas de detecção obtidas, perder-se-á considerar que os resultados obtidos por ambos os classificadores foram bons.

Na secção anterior foram referidos alguns trabalhos que utilizavam técnicas de extracção de arestas para identificar veículos. Este mesmo procedimento poderá ser realizado com base em *haar features*. Em [Yong, X., et al, 2011] propôs-se combinar *haar features* e uma técnica denominada por *Pairwise Geometrical Histograms* (PGH) [Coupe, S., 2009]. O algoritmo consistia em duas etapas:

1. Detectar a região do veículo utilizando um classificador treinado com *haar features*.
2. Com base na região identificada na etapa anterior, calcular o histograma PGH do contorno (*contour*) e comparar com um já existente de um veículo.

Deste modo a primeira etapa deverá detectar o veículo e a segunda deverá reduzir o número de falsos positivos.

Tabela 3. 4 - Comparação dos resultados obtidos na identificação de veículos utilizando *Haar features* e *Haar features* + PGH [Yong, X., et al, 2011].

	<i>Haar Features</i>	<i>Haar Features</i> + PGH
Taxa de detecção	91.25%	95.20%
Taxa de erro	1.68%	1.55%

Os resultados obtidos são ilustrados na tabela 3.4. Verifica-se que utilizando apenas as *haar features*, o algoritmo já obtém bons resultados. Combinando as duas técnicas os resultados melhoram ligeiramente.

Em [Han, S., et al, 2009] é proposto um algoritmo capaz de reconhecer veículos com base em duas características: sombra e arestas verticais. Para identificar a primeira característica é manipulado o brilho da imagem de modo a melhor identificar a sombra do veículo. Já para a segunda característica são utilizadas as *haar features*, que deverão ser capazes de identificar as extremidades laterais do veículo. Com base nestas técnicas são geradas regiões contendo possíveis veículos. De seguida procede-se à validação destas regiões através da identificação de arestas horizontais (com as *haar-features*) e da simetria das arestas detectadas.

Tabela 3. 5 - Resultados da identificação de veículos do algoritmo de Han et al [Han, S., et al, 2009].

	Mesma faixa de rodagem	Diferente faixa de rodagem
Taxa de detecção	91.2%	76,2%

Para os testes foram utilizadas imagens capturadas a partir de um veículo e os resultados são apresentados na tabela 3.5, diferenciando-se os veículos identificados consoante a faixa de rodagem em que se encontravam. A diferença dos resultados ficou-se a dever às regras de simetria que foram impostas, dificultando a detecção de veículos que não se encontrem em posição frontal. Pelo mesmo motivo, o algoritmo apenas é capaz de reconhecer veículos cuja secção traseira é totalmente visível.

Outro trabalho que se dedicou ao reconhecimento de veículos foi desenvolvido por Hota [Hota, R., et al, 2010]. Nesta abordagem foram comparados dois classificadores diferentes: *Adaboost* utilizando *haar features* e SVM combinado com histogramas de gradientes orientados (*Histogram of Oriented Gradi-*

ents). Adicionalmente pretendeu-se testar a detecção a partir de dois ângulos diferentes: traseira e lateral do veículo, sendo que para o treino dos classificadores foram utilizadas, respectivamente, 500 e 550 imagens.

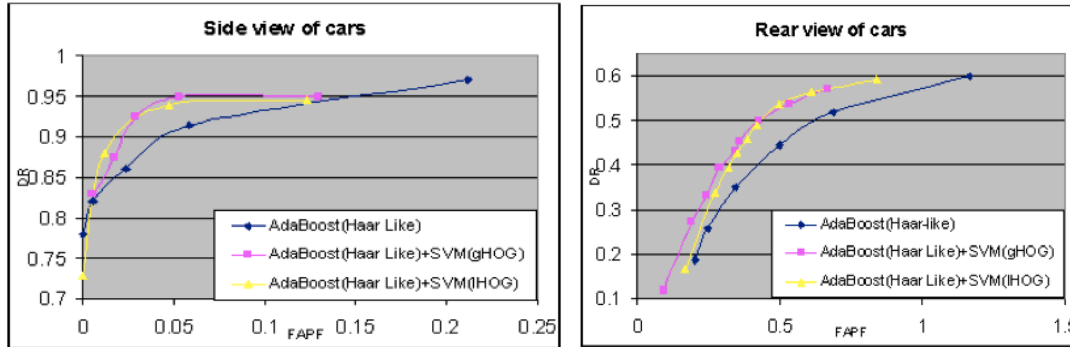


Figura 3. 3 - Resultados da detecção lateral e traseira de veículos utilizando diferentes classificadores [Hota, R., et al, 2010].

Para os testes foram utilizadas 170 imagens contendo 200 veículos. Adicionalmente os autores decidiram testar a eficiência das HOG *features*, diferenciando as calculadas localmente (lHOG) das globalmente (gHOG). Os resultados apresentados na figura 3.3 distinguem a identificação da secção traseira do veículo da lateral.

Analisando a detecção a partir da lateral do veículo, verifica-se que a combinação dos classificadores *Adaboost* e SVM permite a obtenção de melhores resultados, especialmente se considerarmos a quantidade de falsos positivos. No entanto, a utilização singular do *Adaboost* possibilita uma taxa de detecção superior. Em relação ao reconhecimento da traseira do veículo, ambos os classificadores apresentam taxas de detecção semelhantes. No entanto a combinação do *Adaboost* com o SVM, permite obter estes resultados com uma menor quantidade de falsos positivos.

No global, se o objectivo for obter uma boa taxa de detecção aliada a um baixo nível de falsos positivos, a utilização de dois classificadores é mais benéfico. Naturalmente que a adopção desta solução afectará negativamente o tempo necessário para identificar um veículo.

Existem outras situações em que as *haar features* poderão ser utilizadas. Em [Rai, H., et al, 2011] foi apresentado um sistema capaz de localizar consumidores num centro comercial através do reconhecimento dos carrinhos de compra. O processo de detecção consistia na utilização de dois classificadores (*Adaboost* + *haar features* e SVM + HOG *features*) e pretendia-se identificar um código de barras presente nestes objectos. Para o treino dos classificadores foram utili-

zadas 1500 imagens de códigos de barra, das quais 500 se encontravam em posição horizontal e as restantes com diferentes orientações.

Os resultados variam consoante a posição do carrinho, sendo que nos melhores cenários (posição horizontal) são obtidas taxas de detecção de 67%, enquanto nas piores condições (rotação de 45°) estes valores diminuem para 21%

Por fim em [Monteiro, G., *et al*, 2006] foi desenvolvido um sistema de identificação de pedestres utilizando um classificador *Adaboost* com *haar features*. O processo de treino foi composto por 25 etapas, tendo sido utilizadas 4686 imagens positivas e 13934 negativas. Devido a este alargado número de imagens o classificador foi capaz de identificar pedestres em diferentes poses e com diferentes roupas.

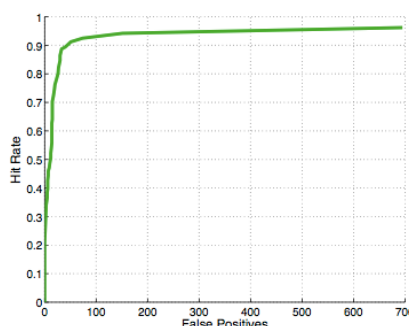


Figura 3. 4 - Curva ROC do sistema de identificação de pedestres [Monteiro, G., *et al*, 2006].

Para os testes do sistema foram utilizadas 1032 imagens de dimensão 213x160, contendo um total de 1588 pedestres. Os resultados apresentados na figura 3.4 permitem verificar que para uma taxa de detecção de 90%, a taxa de identificações incorrectas era apenas 7%.

3.3 Identificação de postes eléctricos

A monitorização e inspecção das redes de energia eléctrica é uma temática amplamente explorada. Em concreto, o foco tem-se centrado nas linhas aéreas existindo actualmente diversos sistemas capazes de auxiliar neste objectivo de inspecção, como o PLMI [Albatroz, 2012]. Recentemente tem sido explorada a hipótese de utilizar veículos não tripulados [Wang, B., *et al*, 2010] [Liu, Y., *et al*, 2012], dispensando deste modo os actuais helicópteros.



Figura 3. 5 - Modelo de poste utilizado por Golightly *et al.*

Apesar destes avanços registados na monitorização das linhas de tensão, poucos esforços têm sido endereçados à detecção automática dos postes eléctricos. Uma das primeiras abordagens [Golightly, I., Jones., D., 2003] data de 2003 e consistia em duas fases:

- Utilização de um *corner detector* [Cooper, J., *et al*, 1991] para determinar pontos de interesse.
- Reconhecimento da região superior do poste com base nos cantos identificados na fase anterior.

Para os testes foi utilizado o modelo de poste apresentado na figura 3.5, abrangendo 4 cenários distintos. Os resultados são apresentados na tabela 3.6, verificando-se que o algoritmo é capaz de identificar o poste com uma boa taxa de sucesso (80% - 90%). Apesar dos bons resultados, os testes foram executados num ambiente controlado, enfrentando algumas dificuldades em cenários contendo vegetação ou variações de luminosidade.

Tabela 3. 6 - Resultados obtidos por Golightly *et al* [Golightly, I., Jones., D., 2003].

	Cenário A	Cenário B	Cenário C	Cenário D
Taxa de Sucesso	90%	88%	80%	89%
Taxa de Erro	24%	20%	27%	17%

Em 2008, Cheng e Song apresentaram uma nova abordagem [Cheng, W., Song, Z., 2008] para a detecção de postes eléctricos. Numa primeira fase são extraídas as arestas na imagem, sendo as mesmas filtradas segundo o comprimento e ângulo de intersecção, com o intuito de delimitar a região do poste. Posteriormente é efectuada a segmentação da imagem utilizando grafos – *Graph Cut*.

O modelo de poste utilizado é idêntico ao da figura 3.5, sendo que para os testes foram utilizados grupos de imagens com diferentes cenários e níveis de luminosidade. Infelizmente não foram apresentados dados sobre os resultados obtidos, sendo apenas enaltecidas a eficiência e precisão do algoritmo. No entanto, tal como no trabalho desenvolvido por Golightly *et al.*, é referida a dificuldade de detecção de postes em cenários complexos.

Um dos trabalhos mais recentes nesta temática foi apresentado em 2010. Desenvolvido por Tilawat *et al* [Tilawat, J., *et al*, 2010] consistia na extracção de aresta utilizando um filtro IIR (*Infinite Impulse Response* [Sarkar, S., Boyer, K., 1991]) e a transformada de *Hough*. Concluída esta fase a imagem é dividida em várias janelas não sobrepostas, sendo contabilizadas o número de arestas contidas em cada uma. A região do poste é determinada dividindo o número de arestas em cada janela pelo máximo da imagem, de modo a normalizar para o intervalo [0, 1].

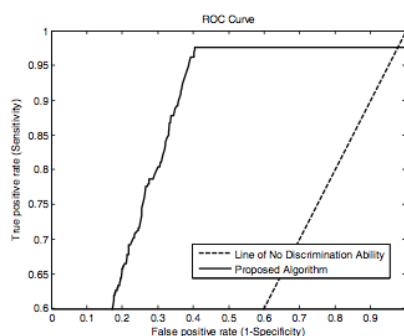


Figura 3. 6 - Resultados do algoritmo de detecção de postes eléctricos proposto por Tilawat *et al* [Tilawat, J., *et al*, 2010].

Para os testes foram utilizadas 50 *frames* aleatórias de um ficheiro de vídeo, sendo os resultados apresentados na figura 3.6. Verifica-se que o desempenho do algoritmo depende da sua configuração. Na sua forma óptima atingiu uma taxa de detecção de 97%, com cerca de 40% de identificações incorrectas. Naturalmente, a redução de falsos positivos afecta negativamente a taxa de detecção que poderá diminuir para 60% com 18% de identificações incorrectas.

Uma das principais vantagens desta abordagem reside na amplitude de postes passíveis de serem detectados. No entanto ao não existir qualquer restrição às arestas identificadas, o número de falsos positivos poder-se-á revelar significativo.

3.4 Protótipos existentes

Actualmente existem dois trabalhos que abordam a temática desta tese - o primeiro data de 2010 e foi desenvolvido por Liu [Liu, W., 2010] e o segundo de 2012 por João Fernandes [Fernandes, J., 2012], ambos em colaboração com a Albatroz Engenharia.

No protótipo desenvolvido por Liu cada *frame* do vídeo era extraída em intervalos de 10, sendo analisada individualmente. O processo de identificação do poste eléctrico consistia nas seguintes etapas:

1. Converter a imagem de RGB para HSV.
2. Filtrar a possível região do poste (região de interesse) com base na cor. Pixéis que não respeitem uma condição previamente definida, serão considerados paisagem e consequentemente descartados.
3. Extrair as arestas existentes na região de interesse, utilizando o operador de *Prewitt* [Maini, R., Aggarwal, H., 2009] e a transformada de *Hough*.
4. Identificar um poste se existirem pelo menos duas arestas diagonais de ângulo $[\pi/4; 3\pi/4]$ que intersectem uma aresta horizontal de ângulo $[3\pi/4; 5\pi/4]$.

Uma vez identificado o poste, procede-se à detecção de possíveis ninhos de cegonha. Nesta fase é assumido que apenas poderão existir ninhos nas regiões assinaladas na figura 3.7.

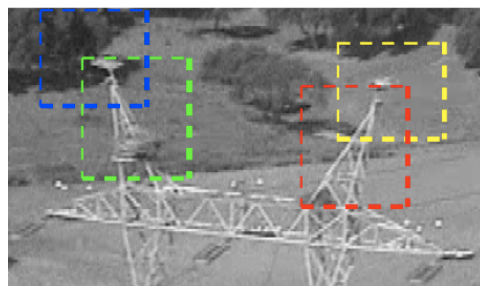


Figura 3. 7 - Identificação das regiões do poste onde poderão existir ninhos [Liu, W., 2010].

A imagem contendo o poste é convertida para *gray-scale* e para cada uma das quatro regiões, os pixéis são analisados individualmente e descartados se a sua cor for inferior a um dado *threshold*. Os pixéis que se encontrem interligados se-

rão considerados como ninhos candidatos, associando aos mesmos uma elipse. Por fim apenas serão considerados como sendo ninhos aqueles cujas elipses formem um ângulo α com a horizontal e possuam uma dimensão superior a x .

Para os testes foram utilizados três ficheiros de vídeo de pequena dimensão. Os resultados obtidos podem ser consultados na tabela 3.7.

Tabela 3. 7 - Resultados obtidos por Liu na identificação de postes eléctricos e ninhos de cegonha

	<i>Frames com</i> Postes	Postes Contabilizados	<i>Frames com</i> Ninhos	Ninhos Contabilizados
Vídeo 1	9	0	-	-
Vídeo 2	7	7	0	1
Vídeo 3	18	17	18	11

Devido à pequena dimensão dos ficheiros de vídeo, o número de *frames* extraídas é reduzido. Acresce que estas mesmas imagens correspondem ao mesmo poste, pelo que os resultados não serão conclusivos. Não obstante, foram analisadas 34 imagens contendo postes eléctricos dos quais 24 foram identificados, correspondendo uma taxa de identificação de 70%. Já para a detecção de ninhos, existiam 18 imagens contendo um ninho tendo sido reconhecidos 11, o que equivale a uma taxa de detecção de 61%. Por fim uma nota para os falsos positivos: de entre as 24 imagens utilizadas no reconhecimento de ninhos, apenas 1 foi incorrectamente identificado.

Em 2012 João Fernandes desenvolveu um novo protótipo relacionado com esta temática [Fernandes, J., 2012]. O algoritmo consistia em duas fases: na primeira seriam identificados os ninhos de cegonha nos postes eléctricos individualmente em cada *frame* do vídeo. Na segunda fase seria explorada a correlação temporal entre as *frames* como forma de melhorar os resultados.

O processo de identificação dos postes eléctricos realizada na primeira fase era composto pelas seguintes etapas:

1. Detecção e extracção das arestas utilizando o *Canny edge detector* e a transformada de *Hough*.

2. Identificação de possíveis áreas do poste. Para cada aresta horizontal que intersecte uma aresta vertical, é definida uma região rectangular de largura w e altura h .
3. Processo de votação entre as possíveis regiões do poste. O ponto anterior é repetido adaptando o valor do *threshold* e são atribuídos pesos consoante a semelhança das arestas extraídas e o modelo do poste.
4. Reconhecimento de um poste. Existindo uma região com peso superior a 65%, considera-se que existe um poste e as 6 arestas que melhor o definem são armazenadas.

Uma vez analisadas todas as *frames* procede-se à segunda fase, cujo objectivo é de melhorar os resultados obtidos na detecção de postes eléctricos. Desde modo considerando a região do poste identificado na *frame n* ampliada em 10% (área de confirmação), procede-se ao processamento da *frame n+1* podendo ocorrer um dos seguintes cenários:

- Na área de confirmação da *frame n+1* existe um poste: Poste identificado na *frame n* é validado.
- Existe um poste na *frame n+1*, mas não se encontra na área de confirmação: Repete-se os passos 3 e 4 da fase anterior na área de confirmação da *frame n+1*. Confirmando-se a existência de um poste, a identificação realizada na *frame n* é validada.
- Não existe um poste na *frame n+1*: Os passos 3 e 4 da fase anterior são aplicados e se for identificado um poste, a *frame n* é validada.

Outro aspecto susceptível a melhoramentos consiste na configuração do poste identificado em cada *frame*. Considerando o poste reconhecido na primeira fase que obteve a melhor classificação na fase de votação, a sua configuração é comparada com a obtida nas *frames n+1* e *frame n-1* e posteriormente analisada a posição do poste na imagem. Se a nova configuração for superior, a anterior é descartada. O processo repete-se pelo conjunto de *frames* existentes.

Uma vez identificado correctamente um poste, procede-se à detecção de ninhos de cegonhas com base na sua cor, forma e posição. Utilizando os modelos de cor RGB e HSV, foi definido um *threshold* que possibilitaria diferenciar os ninhos da paisagem. Deste modo, dada uma imagem, todos os seus píxeis são analisados e catalogados como pertencendo a um ninho se os valores de R, G, B

e V forem inferiores ao respectivo *threshold* e o valor de H for superior. De seguida são conectados os pixéis cuja distância entre si for inferior a 5, considerando esta região a candidata a ninho se o seu comprimento estiver contido num intervalo definido com base numa das arestas do poste. Por fim a cada uma das regiões candidatas é associada uma elipse e se a sua excentricidade pertencer ao intervalo [0.5; 0.9] e esta região estiver contida na área de interesse do poste, é definida a existência de um ninho.

Tal como para a identificação dos postes, a detecção de um ninho é confirmada através da correlação temporal entre *frames*. Considerando a posição do ninho em relação ao poste, é verificada a sua existência na *frame* seguinte.

Tabela 3. 8 - Resultados da detecção de postes e ninhos do algoritmo de João Fernandes.

Identificador do vídeo	Nº de postes	Taxa de detecção	Taxa de FP	Nº de Postes com ninhos	Taxa de detecção	Taxa de FP
1	42	88%	14%	21	83%	33%
2	48	90%	6%	4	100%	18%
3	6	50%	0%	0	-	0%
4	5	40%	33%	0	-	0%
5	64	81%	10%	14	67%	68%
Total	165	83%	14%	39	79%	43%

Actualmente existem dois conjuntos de testes para este algoritmo. No primeiro conjunto, apresentado na tabela 3.8, foram utilizados 5 sequências de vídeo contendo um total de 165 postes eléctricos e 39 ninhos de cegonhas. Verifica-se que 83% dos postes existentes foram correctamente identificados, existindo apenas 14% de identificações incorrectas. Quando comparado com os resultados obtidos por Liu, denota-se uma melhoria de sensivelmente 13% na taxa de detecção.

Em relação à detecção de ninhos, os dados ilustrados na tabela 3.8 referem-se a postes eléctricos em risco – postes que contêm pelo menos um ninho. Assim sendo não será possível relacionar directamente a quantidade de ninhos existentes com o número de ninhos correctamente identificados. Não obstante, 79% dos postes contendo pelo menos um ninho foram correctamente detectados. Por

outro lado, 43% dos postes foram identificados como estando em risco, quando na realidade não continham qualquer ninho.

O segundo conjunto de testes é apresentado nas figuras 3.8 e 3.9, encontrando-se separados os resultados da identificação de postes eléctricos da detecção de ninhos de cegonha.

Seq. #	Nr. of Frames	Tower HR		% detect towers with accept. config.	
		Still images	Video seq.	Before mot. an.	After mot. an.
1	41	39%	73%	23%	37%
2	7 (4 with tower)	75%	100%	25%	50%
3	24	54%	63%	53%	100%
4	81 (70 with tower)	17%	80%	30%	64%
5	34	24%	76%	50%	88%
6	15 (14 with tower)	79%	100%	50%	86%
7	11	73%	91%	50%	80%
8	48	33%	67%	38%	81%
9*	41	10%	100%	54%	78%
10*	37	57%	100%	76%	97%
11	18	28%	94%	53%	59%
12	12	8%	100%	33%	100%
13	55	25%	78%	42%	79%
14	11	36%	82%	67%	78%
15	64	19%	45%	24%	52%
16*	41	22%	100%	59%	95%
17	47	60%	98%	70%	100%
18	49	59%	100%	43%	78%
19*	71	11%	34%	54%	100%
20	7	86%	100%	57%	71%
Global	699 with tower	33%	78%	48%	80%

Figura 3. 8 - Resultados da identificação de postes eléctricos por João Fernandes [Fernandes, J., 2012].

Como dados para os testes foram utilizadas 20 sequências de vídeo de pequena duração, cada uma destas sequências dedicadas a um único poste. De entre os postes existentes 13 continham pelo menos 1 ninho, existindo no total 23 ninhos de cegonha.

Na figura 3.8 são apresentados os resultados para a identificação de postes eléctricos separados em três categorias – imagens, vídeo e por fim análise de vídeo considerando o posicionamento do poste na *frame* (*motion vector*). As vantagens da utilização da correlação temporal entre *frames* fica patente na diferença de resultados entre imagens e vídeo. Descurando a relação entre *frames*, de entre as 699 imagens contendo um poste, apenas 33% destes foram correctamente identificados. Utilizando todas as potencialidades do algoritmo, foi possível obter uma taxa de detecção de 80%.

Terminada a identificação dos postes eléctricos, procedeu-se à detecção dos ninhos de cegonha. Os resultados apresentados na figura 3.9 encontram-se divididos em análises de imagens singulares e em vídeo utilizando a correlação temporal entre *frames*. Novamente existe uma grande discrepância entre as duas técnicas. Analisando as imagens individualmente, verificou-se que apenas 40%

dos ninhos existentes foram identificados, existindo ainda 37% de identificação incorrectas. Por sua vez relacionando as *frames*, a taxa de detecção aumentou para 78% e a quantidade de identificações incorrectas diminui para 22%.

Seq. #	Still images analysis				Video sequences analysis			
	Frames	Nests/frame	Nest HR	Nest FPR	Existing nests [L H R]	Detected nests [L H R]		
1	41	1	5%	60%	[1 0 0]	[0 0 0]		
2	7	0	-	7	[0 0 0]	[0 0 0]		
3	15	0	-	29	[0 0 0]	[0 0 0]		
4	81	0	-	94	[0 0 0]	[0 1 0]		
5	34	0	-	19	[0 0 0]	[0 0 0]		
6	15	0	-	11	[0 0 0]	[0 0 0]		
7*	11	2	28%	0%	[1 0 1]	[1 0 1]		
8	41	0	-	118	[0 0 0]	[0 0 1]		
9*	41	2	68%	57%	[1 0 1]	[1 0 1]		
10	37	4	68%	16%	[2 0 2]	[2 0 1]		
11	18	2	72%	37%	[1 0 1]	[1 0 1]		
12	12	2	88%	16%	[1 0 1]	[1 1 1]		
13	55	1	18%	57%	[1 0 0]	[0 0 0]		
14*	11	3	76%	14%	[1 0 2]	[1 1 2]		
15	63	2	19%	40%	[1 0 1]	[1 0 0]		
16	41	0	-	5	[0 0 0]	[0 0 0]		
17*	47	1	17%	60%	[1 0 0]	[1 0 0]		
18	51	1	6%	80%	[0 0 1]	[0 0 0]		
19	71	1	24%	0%	[0 0 1]	[0 0 1]		
20	7	1	29%	60%	[1 0 0]	[1 0 1]		
Overall	699	743	40%	37%	Nest HR = 78%		Nest FPR = 22%	

Figura 3. 9 - Resultados da detecção de ninhos de cegonha por João Fernandes [Fernandes, J., 2012].

Considerando a mesma metodologia utilizada no primeiro conjunto de testes (tabela 3.8) para a detecção de ninhos, 77% dos postes contendo pelo menos um ninho foram correctamente identificados. Destes em 50% foram detectados a totalidade dos ninhos existentes.

Comparando os resultados obtidos por Liu e por João Fernandes, verifica-se que a abordagem deste último possibilita ganhos na taxa de identificação de postes em cerca de 10% e na detecção de ninhos de sensivelmente 18%.

4

Solução Proposta

Neste capítulo é efectuada uma descrição detalhada do protótipo desenvolvido para a detecção de ninhos de cegonha em postes eléctricos.

Inicialmente é apresentada uma visão global do funcionamento do algoritmo proposto. De seguida cada uma das seguintes etapas é aprofundada na respectiva secção:

- Análise de ficheiros de vídeo
- Pré-Processamento de imagem
- Identificação de postes eléctricos
- Detecção de ninhos de cegonha

O algoritmo proposto foi desenvolvido em *c++*, utilizando a biblioteca *Open Source Computer Vision Library (OpenCV)* na versão 2.4.2.

4.1 Visão geral

No capítulo anterior foram apresentados os dois trabalhos existentes para a detecção de ninhos de cegonha. No trabalho desenvolvido por Liu as *frames* eram extraídas do vídeo em intervalos regulares e processadas individualmente. Já João Fernandes optou por estender esta abordagem, fazendo uso da correlação temporal entre *frames* como meio de melhorar os resultados de detecção. No âmbito desta tese o foco residiu na análise individual de imagens, exploran-

do diferentes técnicas de processamento de imagem com o intuito de obter os melhores resultados possíveis. Para além da vantagem natural de possibilitar uma detecção eficaz em imagens singulares (quando comparado com o trabalho desenvolvido por João Fernandes), permite ganhos de performance uma vez que não será necessário o processamento da totalidade das *frames*. Adicionalmente é menos susceptível a possíveis movimentações bruscas do helicóptero. Considerando que os vídeos são capturados tipicamente com uma *frame-rate* entre 25 e 30, extraíndo as *frames* em intervalos de 10 permite que para um dado poste existam em média 10 imagens passíveis de serem analisadas. Este conjunto de imagens disponibiliza uma amostra de dados em diferentes escalas e serão suficientes para determinar a existência de postes e ninhos.

Uma vez extraída uma *frame* são desencadeados um conjunto de operações responsáveis para inicialmente identificar a região do poste e de seguida a localização dos ninhos existentes. Na figura 4.1 é apresentada uma visão geral do funcionamento do algoritmo.

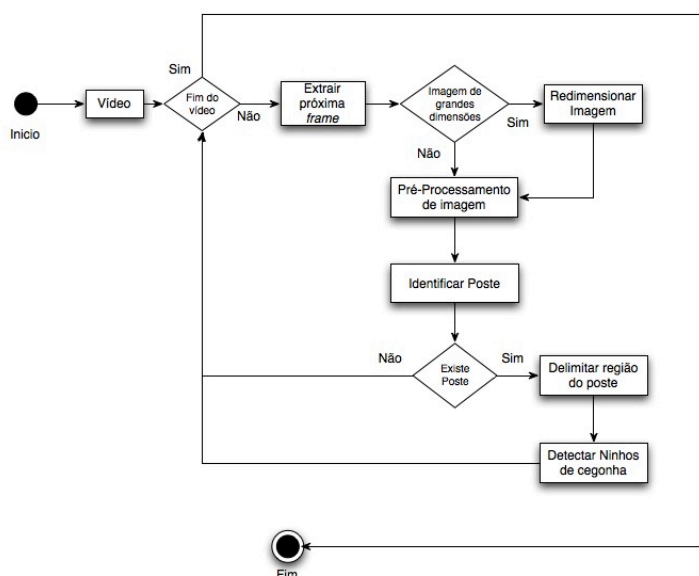


Figura 4. 1 - Visão geral do funcionamento do algoritmo.

A primeira etapa consiste em analisar as dimensões da imagem. Se a largura exceder os 1200 pixéis, esta será redimensionada com um factor de escala que permita obter a largura referida. De seguida são efectuadas três cópias à imagem original e aplicadas diferentes técnicas a cada uma.

Para o reconhecimento do poste, são utilizados o *Canny edge detector* e a transformada de *Hough* que permitem a extracção das arestas contidas da imagem. Considerando as dimensões da imagem, o objectivo consiste em recolher entre 800 a 1000 arestas. Esta etapa ocorre de forma cíclica, onde em cada iteração o valor de *threshold* é adaptado automaticamente até se atingir o número de arestas pretendido. O passo seguinte consiste em filtrar as arestas extraídas segundo um conjunto de restrições que permitem definir um poste eléctrico. Utilizando as arestas remanescentes, deverá se possível identificar as extremidades do poste e por conseguinte será possível delinear uma região em que o mesmo esteja inserido.

O procedimento anterior é repetido para cada uma das quatro imagens, os resultados obtidos são analisados e existindo mais que uma identificação é delimitada a região que melhor define o poste.

Uma vez identificado o poste, procede-se à detecção de possíveis ninhos contidos no mesmo. Previamente havia sido treinado um classificador utilizando *haar-features* com um conjunto alargado de imagens positivas (contendo ninhos) e imagens negativas (sem ninhos). Aplicando o classificador à imagem e restringindo a região de pesquisa ao poste, será possível localizar a posição e a dimensão dos ninhos na imagem. Este processo deverá ser capaz de detectar a maioria dos ninhos existentes, com a vantagem de não se cingir a determinadas regiões do poste.

Terminada detecção de ninhos na imagem, procede-se à análise das restantes *frames* do vídeo.

4.2 Análise de ficheiros vídeo

Como já foi referido a análise de ficheiros de vídeo desenrola-se através da extracção de *frames* em intervalos de 10 e posterior processamento individual. A apreciação de todas as *frames* originaria custos de computação desnecessários, uma vez que a diferença entre imagens consecutivas, tipicamente, é reduzida. Desde modo se o algoritmo não fosse capaz de reconhecer um poste ou ninho na *frame n*, provavelmente seria igualmente incapaz de o fazer na *frame n+1*. Considerando que para cada poste/ninho existirão em média 10 imagens passíveis de serem analisadas, uma correcta identificação em duas destas imagens seria suficiente para garantir, com algum grau de certeza, a existência do objecto pretendido. Outra vantagem do processamento intervalado de *frames* reside

na menor probabilidade de falsos positivos. Por vezes encontram-se nos cenários padrões que partilham algumas semelhanças com os objectos que se pretende reconhecer, originando identificações incorrectas. Nestas situações, uma análise sucessiva das *frames* apenas acentuaria o problema.

O algoritmo para o processamento do vídeo será o seguinte:

```
Array < pylonNumber, numberOfNests> finalResults
Array nests;

while (currentFrame <= totalNumberOfFrames){

    //Get next frame
    frame = getFrame(video, currentFrame)

    // Resize frame if width superior to 1200
    if (frame.width > maxWidth)
        resize(frame);

    //Search for pylon on frame
    hasPylon = findPylon (frame);

    if (hasPylon){

        // Find nest on region surrounding the pylon
        numberOfNests = findNests (frame);

        // Add nest count if gap between frame not containing pylon is less than 2
        if (framesWithoutPylon() <= 2)
            nests.put(numberOfNests);

        // New pylon found. Save results from previous counting considering the
        // median number of nests
        else{
            if (nests.size() > 2){
                finalResults.put(pylonNumber, Median(nests));
                pylonNumber++;
            }

            nests.clean();
            nests.put(numberOfNests);
            clearFramesWithoutPylon();
        }
    }

    // No pylon found. Increment count of consecutive frames without pylon
    else
        incrementFramesWithoutPylon();

    // Increment frame count by factor 10
    currentFrame += 10;
}
```

Uma vez extraída a *frame* procede-se à identificação do poste e posterior detecção de ninhos. Nesta fase poderá ocorrer um dos seguintes cenários:

- Nenhum poste é identificado: Incrementa-se um contador de *frames* sucessivas sem postes eléctricos.
- Um poste é identificado e o número de *frames* sucessivas sem poste é inferior a 3: o número de ninhos detectados é armazenado.
- O poste é identificado, mas existiram 3 ou mais *frames* consecutivas sem qualquer poste: Considera-se o poste identificado como sendo novo, sendo necessário processar a informação referente ao poste anterior. Se existiram pelo menos duas identificações, escolhe-se a mediana do número de ninhos detectados, associando este valor a um identificador único do poste. No caso de ter existido apenas um reconhecimento, considera-se que o mesmo terá sido um falso positivo.

As anteriores condições permitem melhorar os resultados obtidos. Assim se um poste for reconhecido uma única vez este é descartado, sendo apenas apreciados para os resultados finais situações onde existam pelo menos duas identificações. A cada poste é atribuído um identificador, sendo este valor incrementado quando nas três *frames* imediatamente anteriores não tenha sido identificado um poste.

O processo anterior é repetido até que todas as *frames* do vídeo sejam analisadas, altura em que é criado um ficheiro contendo o número de ninhos detectados para cada poste.

Nas próximas secções será explicado em maior detalhe as diferentes operações intermédias do algoritmo.

4.3 Pré Processamento de Imagens

Após a extracção da imagem são verificadas as suas dimensões e se a sua largura exceder os 1200 pixéis a imagem é redimensionada. O principal motivo desta operação prende-se com ganhos de eficiência.

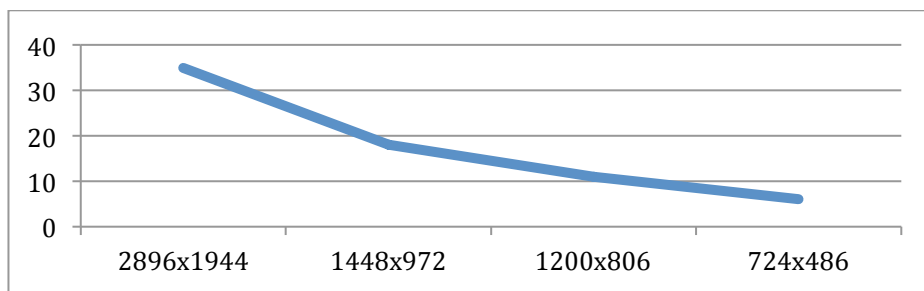


Figura 4. 2 – Variação do tempo (em segundos) necessário para o processo de identificação em função da resolução da imagem.

Na figura 4.2 é apresentada a variação do tempo de execução do algoritmo em função da resolução da imagem. É notória uma redução drástica do tempo à medida que o número de pixéis na imagem diminui. Acresce que apesar das dimensões da imagem diminuírem, a taxa de detecção aumenta ligeiramente. Na figura 4.3 são apresentados os resultados da detecção de ninhos em imagens de diferentes dimensões. De notar que o algoritmo se encontra otimizado para imagens de 1200 pixéis de largura. Não obstante as três resoluções testadas, esta foi a que apresentou um menor número de ninhos falsos positivos. Utilizando imagens de menores proporções existe uma maior dificuldade em extrair as arestas pertencentes ao poste. Adicionalmente a detecção de ninhos de menor dimensão poderia tornar-se mais problemática.

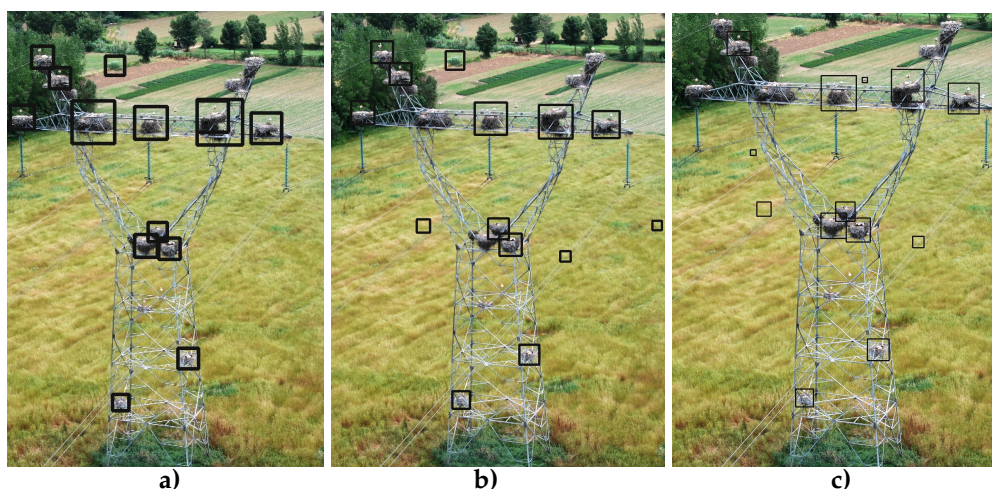


Figura 4. 3 - Resultados da detecção de ninhos utilizando imagens de diferentes resoluções.

Com o intuito de encontrar um compromisso entre o tempo de execução e a qualidade da imagem optou-se por considerar uma largura máxima de 1200 pixéis.

No passo seguinte são criadas três cópias à imagem original e aplicadas as seguintes transformações:

1. A imagem original é convertida para *gray scale*.
2. A primeira cópia é convertida para *gray scale* e de seguida é efectuada uma equalização do histograma (*histogram equalization*) e uma suavização gaussiana.
3. Na segunda cópia é efectuada uma conversão do modelo de cores RGB para HSV. De seguida os *pixéis* são filtrados segundo um *threshold* definido para o *Value*.
4. Na terceira cópia o procedimento aplicado no ponto anterior é repetido sendo posteriormente efectuada uma equalização do histograma e uma suavização gaussiana.

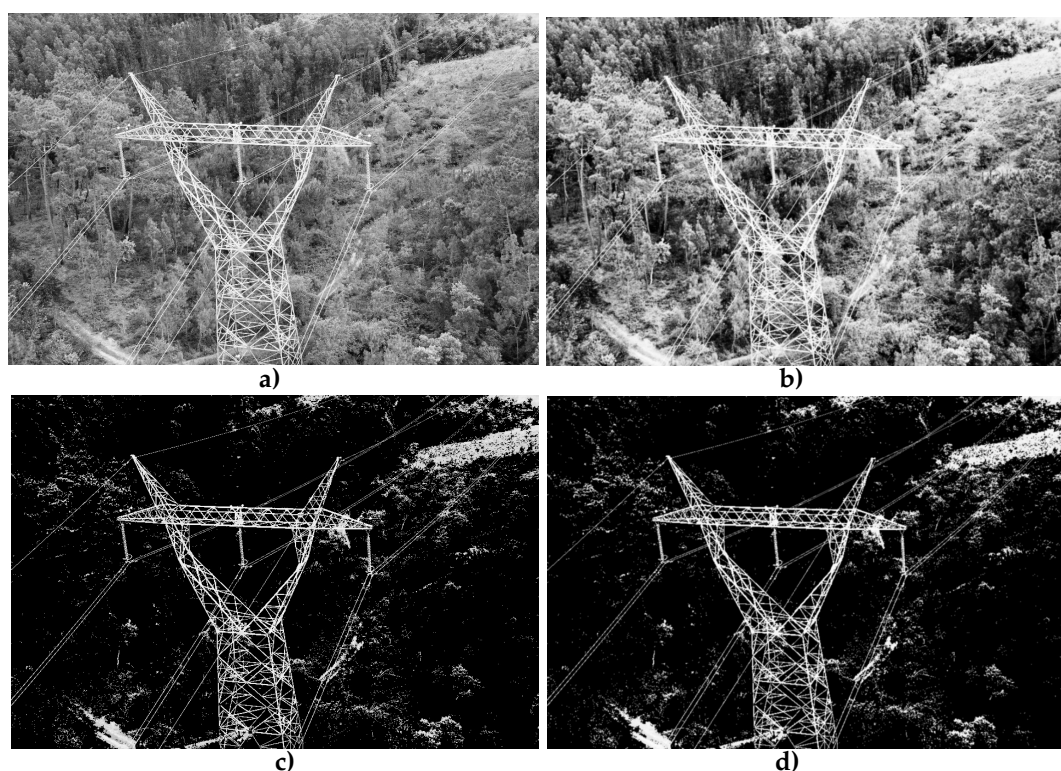


Figura 4. 4 - Imagens resultantes após transformações. a) Primeira transformação, b) Segunda transformação, c) Terceira transformação, d) Quarta transformação.

Na figura 4.4 são apresentadas as imagens resultantes das transformações aplicadas. O objectivo de utilizar este conjunto de imagens reside na maior probabilidade de identificar um poste. Um não reconhecimento invalidaria o restante algoritmo. A equalização do histograma permite ampliar o contraste da ima-

gem em locais onde este valor é reduzido, o que poderá aumentar a capacidade do algoritmo de identificar o poste. No entanto esta operação poderá intensificar o ruído presente na paisagem, razão pela qual à posteriori é efectuada uma suavização gaussiana.

Outro método utilizado consiste em filtrar os pixéis da imagem com base na sua cor. Um poste tipicamente é reconhecido pelos seus tons de cinzento, sendo essa a cor escolhida como *threshold*. De modo a melhor expressar uma condição que defina a cor pretendida, a imagem original é convertida para o modelo de cores HSV. De entre as três componentes, que permitem valores compreendidos entre 0 e 255, o de interesse é o *Value* tendo sido definido um *threshold* de 205. Assim sendo, são analisados todos os pixéis da imagem e aqueles que possuírem um *value* inferior a 205 são convertidos para preto. Os restantes mantêm a sua cor, que deverá apresentar diferentes tonalidades de cinzento. Através da figura 4.4 c) é possível verificar que este método foi capaz de reduzir o impacto da paisagem, sem contudo perder o detalhe do poste.

Deve ser mencionado que apesar de a maior dos postes apresentarem a cor referida, existem algumas variantes que se encontram pintadas de outras cores, como branco/vermelho. Para estas situações, as transformações 3 e 4 não deverão ser capazes de identificar o poste.

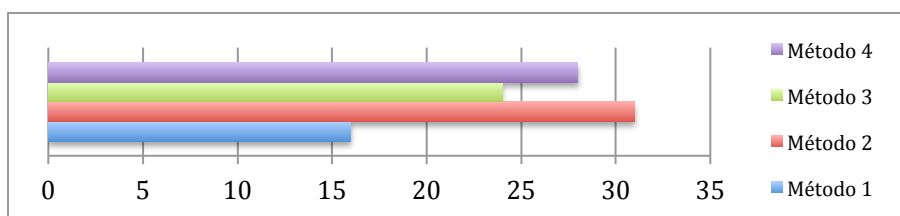


Figura 4. 5 - Resultados da identificação de postes dos 4 métodos propostos. Método 1: Conversão para *gray-scale*. Método 2: Equalização do histograma + suavização gaussiana. Método 3: Filtragem por cor. Método 4: Filtragem por cor + Equalização do histograma + suavização gaussiana.

Os dois métodos apresentados foram conjugados nas quatro imagens, permitindo deste modo cobrir um conjunto variado de cenários.

Na figura 4.5 são apresentados os resultados da identificação de postes para cada um dos quatro métodos. Verifica-se que o segundo método é aquele que obtém a melhor taxa de detecção – 77%, contrastando com o primeiro método que apenas foi capaz de reconhecer 40% dos postes.

A disparidade de resultados foi o principal motivo da utilização dos quatro métodos, uma vez que permite obter uma boa capacidade de identificação de postes, mantendo um baixo nível de falsos positivos através da obrigatoriedade de existirem pelo menos duas detecções.

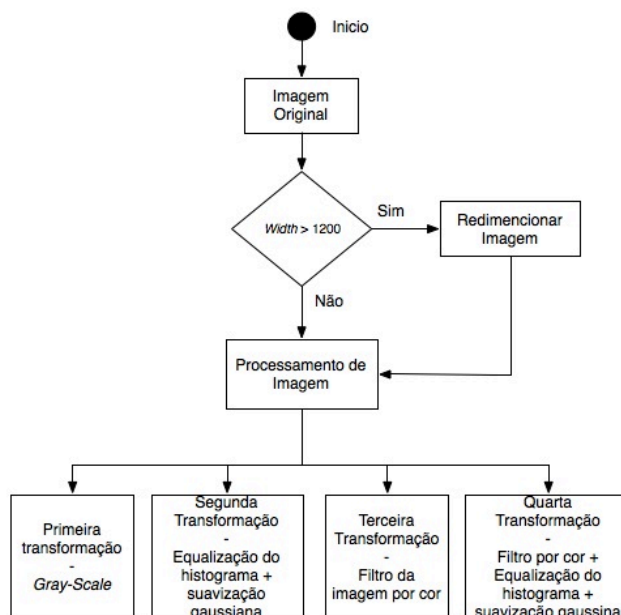


Figura 4. 6 - Esquematização do pré-processamento efectuado.

Na figura 4.6 são ilustradas as diferentes transformações aplicadas à imagem, necessárias na próxima etapa do algoritmo.

Na secção seguinte será apresentado em detalhe a operação de identificação de postes eléctricos.

4.4 Identificação de postes eléctricos

Uma das etapas mais críticas do algoritmo é a referente à identificação do poste eléctrico. Não só é essencial reconhecer a presença de um poste na imagem, como é igualmente importante delimitar a região em que o mesmo se encontra inserido. Na figura 4.7 é esquematizada a sequência de eventos necessários para determinar a existência de um poste.

O primeiro passo consiste em extrair as arestas da imagem. De seguida as arestas são agrupadas consoante a sua orientação, sendo eliminadas as que se encontrem isoladas. O passo seguinte consiste em verificar se as arestas formam

um 'V' característico de um poste. Verificando-se as anteriores condições, são utilizados pontos-chave para determinar a região da imagem em que o poste se encontra inserido.

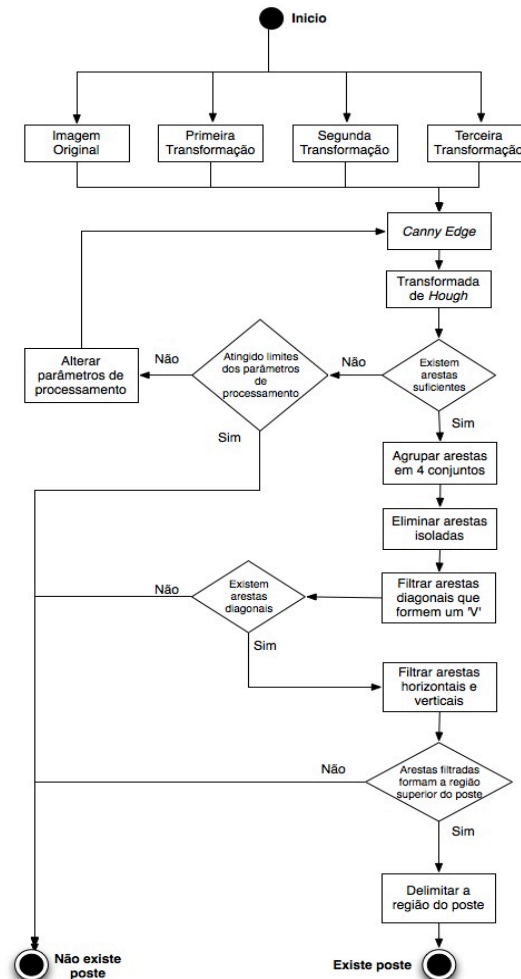


Figura 4. 7 - Encadeamento das diferentes operações efectuadas na identificação de um poste.

Para a extracção das arestas é utilizado o *Canny edge detector* e a transformada de *Hough*. Este processo ocorre de forma cíclica, ou seja, a imagem é analisada diversas vezes até que seja extraído um determinado número de arestas. Um dos maiores problemas na identificação de postes reside na diversidade de cenários em que os mesmos se encontram enquadrados. Existindo cenários mais ou menos complexos, os parâmetros óptimos das técnicas utilizadas para a extracção de arestas variam. Por este motivo uma imagem poderá ser analisada diversas vezes, onde em cada iteração os *thresholds* são adaptados automaticamente. Como condição de aceitação, o número de arestas extraídas deve encontrar-se situado entre 800 e 1000. Este intervalo foi escolhido tendo em conside-

ração as dimensões da imagem, permitindo um conjunto de dados suficiente para reconhecer um poste. Se o número de arestas fosse inferior, corria-se o risco de que alguns elementos importantes do poste não fossem reconhecidos. Por outro lado, se fosse excessivamente superior aumentaria o ruído dificultando a identificação do poste e aumentando a probabilidade de falsos positivos.

O funcionamento desta fase do algoritmo será o seguinte:

```
threshold = 150

while (!(lines > 800 && lines < 1000)){

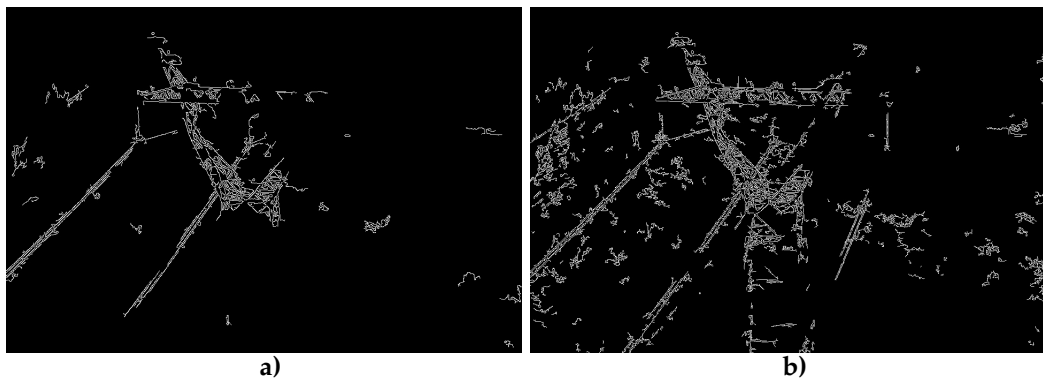
    // Extract lines from image
    Canny(input, output, threshold, threshold*2);
    HoughLines(output, lines);

    // Decrease threshold if the number of lines extracted from the image is less then 800
    if(lines < 800)
        threshold -= (800 - lines)/25;

    // Increase threshold if the number of lines extracted from the image is greater then 1000
    if(lines > 1000)
        threshold += (lines - 1000)/25

}
```

Definido um *threshold* inicial de 150, este valor será alterado em cada alteração até que seja atingido o número de arestas pretendido. Na figura 4.8 encontram-se ilustradas as arestas identificadas nas diferentes iterações. Nas imagens a) e b) verifica-se que o número de arestas presentes são insuficientes para correctamente identificar o poste. Na imagem c) existe demasiada interferência da paisagem, dificultando o processo de reconhecimento. Já na imagem d), mantendo-se alguma interposição da paisagem, existem elementos suficientes do poste para que o mesmo seja detectado.



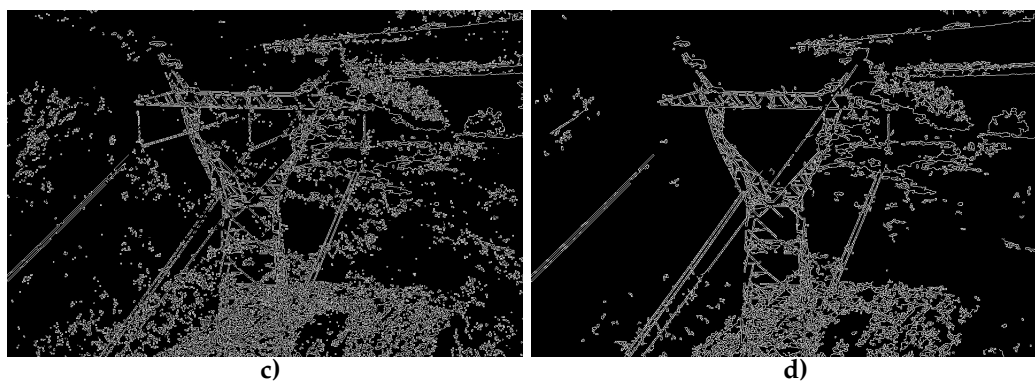


Figura 4. 8 - Arestas identificadas pelo *Canny edge detector* utilizando diferentes *thresholds*. a) e b) Menos de 800 arestas. c) Mais de 1000 arestas. d) Entre 800 e 1000 arestas.

Com base nas arestas identificadas pelo *Canny edge detector*, em concreto as da figura 4.8 d), é possível extrair as mesmas utilizando a transformada *Hough*. Na figura 4.9 encontram-se ilustrados os resultados desta operação. Apesar de diversas arestas realmente pertencerem ao poste, existe um conjunto das mesmas que pertence à paisagem.

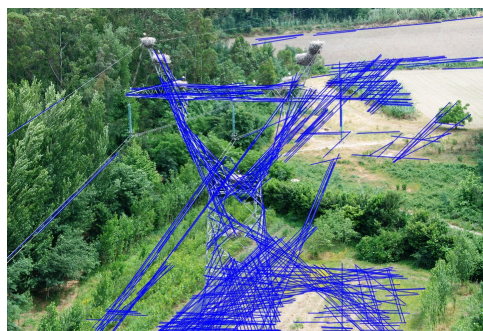


Figura 4. 9 - Conjunto de rectas extraídas pela transformada de *Hough*.

A etapa seguinte do algoritmo consiste na análise e filtragem das arestas extraídas. Inicialmente as rectas são agrupadas nos conjuntos vertical, horizontal, diagonal com ângulo agudo e diagonal com ângulo obtuso. Cada aresta é analisada individualmente e adjudicada a um dos grupos se respeitar uma das seguintes condições:

- Grupo horizontal: ângulo inferior a 15° .
- Grupo vertical: ângulo compreendido entre 80° e 100° .
- Grupo diagonal com ângulo agudo: ângulo superior a 45° e inferior a 80°
- Grupo diagonal com ângulo obtuso: ângulo superior a 100° e inferior a 135° .

Todas as rectas que não apresentem um dos ângulos anteriores são eliminadas. O objectivo é que cada um dos grupos represente uma secção do poste. Na figura 4.10 são apresentadas as regiões do poste a que cada grupo corresponde. Assim sendo as rectas do grupo horizontal encontram-se representadas a verde, as verticais a preto, as diagonais com ângulo agudo a vermelho e por fim as diagonais com ângulo obtuso a azul.



Figura 4. 10 - Identificação das secções do poste consoante os ângulos das suas arestas.

As condições anteriores foram definidas considerando que o poste se encontra na posição frontal. Se tal não se verificar as rectas poderão ser incorrectamente agrupadas, levando a que um poste não seja identificado.

O passo seguinte consiste em eliminar as rectas que se encontrem isoladas. Tipicamente nas regiões que definem o poste são extraídas diversas arestas, encontrando-se as mesmas próximas umas das outras. Quando tal não se verifica, geralmente é porque são pertencentes à paisagem. No âmbito do trabalho desenvolvido, considera-se que uma recta se encontra isolada se não existirem pelo menos outras duas nas proximidades. O conceito de proximidade é calculado com base nas dimensões da imagem. Assim sendo as arestas vizinhas terão que se encontrar a pelo menos x pixéis de distância na horizontal e y na vertical, com:

$$x = \frac{\text{largura da imagem}}{100}$$

$$y = \frac{\text{altura da imagem}}{100}$$

A eliminação destas rectas é importante uma vez que poderiam influenciar negativamente a operação seguinte referente ao reconhecimento da região superior do poste.

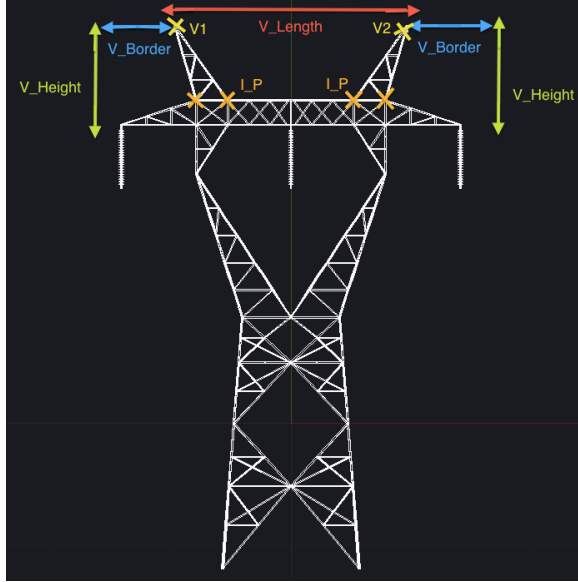
O padrão que se pretende identificar é a forma em 'V', sendo que para tal o foco incidirá nas arestas diagonais. Estas rectas são analisadas e deverão respeitar as seguintes condições:

- Intersectar pelo menos uma aresta horizontal.
- As rectas pertencentes a ambos os grupos das arestas diagonais não poderão intersectar-se.
- As rectas diagonais com ângulo obtuso deverão se encontrar à esquerda das rectas com ângulo agudo.
- A distância horizontal entre rectas dos dois grupos deverá ser superior a $\frac{\text{largura da imagem}}{10}$.

As arestas diagonais que não respeitarem esta condição serão removidas. De forma simplificada esta tarefa é executada da seguinte forma:

```
foreach l in diagonal_left{
    foreach r in diagonal_right{
        // Both diagonal edges must intersect at least one horizontal one
        if (intersect(r, horizontal_lines) && intersect(l, horizontal_lines)){
            // The diagonal edges must not intersect with each other
            if (!intersect(l, r)){
                // Edges with angle superior to 100° must be left to edges with angle inferior to 80°
                if (isLeft(l, r)){
                    // Distance between edges must be superior to 10% of the image width
                    if (distance(l, r) > image.width/10){
                        if (!new_diagonal_left.contains(l))
                            new_diagonal_left.put(l);
                        new_diagonal_right.put(r);
                    }
                }
            }
        }
    }
}
```


Uma vez identificada a região do poste em 'V' é possível modelar o poste, como ilustrado na figura 4.11.



V1 – Vértice Superior Esquerdo

V2 – Vértice Superior Direito

V_{Length} – Distância horizontal entre os vértices superiores

V_{Height} – Distância vertical entre os vértices superiores e as arestas horizontais

V_{Border} – Distância horizontal entre os vértices superiores e as extremidades laterais do poste

I_P – Pontos de intersecção das arestas diagonais com as horizontais

Figura 4. 11 - Dimensões e pontos de interesse do poste.

Assim, com base nas arestas diagonais é possível determinar os pontos $V1$ e $V2$ e por conseguinte V_{Length} . Através desta informação é possível estimar as restantes variáveis segundo:

$$V_{Border} = \frac{V_{Length}}{3}$$

$$V_{Height} = \frac{V_{Length}}{3}$$

Nesta fase já é possível determinar os locais onde é expectável que existam arestas do poste, nomeadamente as horizontais e verticais, definindo as seguintes áreas:

- Horizontal: $x \in [V1 - V_{Border}, V2 + V_{Border}]$ e $y \in [V1 + \frac{V_{Length}}{4}, V1 + V_{Height}]$
- Vertical: $x \in [V1, V2]$ e $y \in [V1 + V_{Length}, image_height]$

Por fim os grupos de arestas verticais e horizontais são processados, sendo apenas considerados os que se inserirem nas áreas definidas anteriormente. Considera-se que existe um poste se existirem um mínimo de 10 arestas horizontais, 10 diagonais com ângulo obtuso, 10 arestas com ângulo agudo e x arestas verticais. O valor de x será atribuído segundo:

- $x = 5$, se $V_{Length} < \frac{image_width}{3}$. A região inferior do poste deverá ser visível na imagem
- $x = 0$, se $V_{Length} > \frac{image_width}{3}$. A região superior do poste ocupa uma grande proporção da imagem, pelo que a zona inferior não deverá ser visível.

Se as condições anteriores se verificarem, o resultado deverá ser semelhante ao apresentado na figura 4.12.

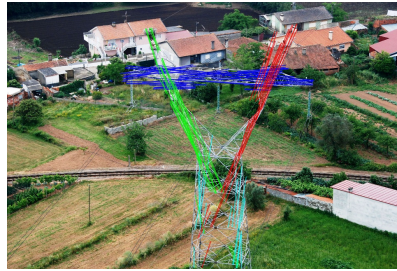


Figura 4. 12 - Resultado do processo de identificação das arestas do poste.

Uma vez identificado, resta apenas delimitar uma área em redor do poste. Conhecendo as suas extremidades, este processo é relativamente trivial.

Como já referido, existem quatro imagens que terão que ser analisadas para identificar um poste. Para que o mesmo seja considerado válido, deverão existir pelo menos 2 reconhecimentos. Quando tal se verifica, a área final que delimita o poste será determinada através da mediana das regiões definidas em cada imagem.

Existindo uma correcta identificação do poste, prossegue-se com a detecção de ninhos de cegonha contidos no mesmo.

4.5 Detecção de ninhos de cegonha

Uma vez identificado um poste, procede-se à detecção de ninhos de cegonhas. A primeira operação consiste no recorte da imagem em redor da região

delimitada na etapa anterior. Este procedimento é importante uma vez que permite:

- Reduzir o número de falsos positivos: Tendo em conta que os ninhos são construídos com detritos existentes nas proximidades, existe o risco que ocorram detecções incorrectas na paisagem. Restringindo a região de pesquisa à área do poste, reduz-se a probabilidade destas ocorrências.
- Aumentar a performance do algoritmo: O processo de detecção de ninhos consiste em definir uma janela de pesquisa, analisando a totalidade da imagem. Deste modo cingindo a região de procura à do poste, consegue-se diminuir o tempo de processamento.

A fase seguinte consiste na detecção dos ninhos de cegonha existentes na imagem recortada. Este processo é obtido através da utilização de um classificador construído com recurso a *haar features*. Previamente este classificador havia sido treinado, utilizando um total de 3321 imagens positivas (contendo ninhos) e 5642 negativas (paisagem). As imagens de ninhos correspondem a recortes efectuados previamente, como ilustrado na figura 4.13. Foram escolhidos ninhos de diferentes dimensões e capturados em diferentes ângulos, com o intuito de abranger um variado conjunto de possibilidades. Já para as imagens negativas, estas são essencialmente constituídas por zonas agrícolas/florestais e urbanas.



Figura 4. 13 - Excerto de algumas imagens utilizadas no treino do classificador de ninhos.

O processo de treino decorreu numa máquina *AMD Athlon 64 X2 Dual Core 4800+ (2.50 GHz)*, com 2GB de memória RAM e sistema operativo *Windows 7*, tendo decorrido durante cerca de 23 dias.

Este classificador foi construído utilizando ferramentas disponíveis no *OpenCV* e definido os seguintes parâmetros:

- Subjanela de dimensão 20x20
- 25 etapas de treino
- *Minimum hit ratio* 0.995 (valores por defeito)
- *Maximum false alarm ratio* 0.50 (valores por defeito)
- 1 GB de memória RAM

Estes valores foram considerados após a realização de testes em pequena escala. Um classificador foi treinado constituído por 20 etapas e contendo 150 imagens positivas e 300 negativas. Verificou-se que nestas condições o algoritmo era capaz de detectar mais de 50% dos ninhos existentes. Para o classificador final optou-se por escalar os parâmetros escolhidos. Optou-se por utilizar um processo de treino composto por 25 etapas, uma vez que permitiria uma melhor capacidade de detecção. No algoritmo de detecção de faces de Jones e Viola foram utilizadas 35 etapas, mas devido a questões temporais esta opção não seria viável. Na figura 4.14 é apresentado um excerto da primeira etapa do processo de treino do classificador, tendo sido utilizadas 125199 *haar features*.

```
Number of features used : 125199
Parent node: NULL
*** 1 cluster ***
POS: 3321 3321 1.000000
NEG: 5642 1
BACKGROUND PROCESSING TIME: 0.11
Precalculation time: 79.72
```

N	zSMP	IF	ST.THR	HR	FA	EXP. ERR
1	100x	-	-0.476200	1.000000	1.000000	0.202164
2	100x	-	-0.768853	1.000000	1.000000	0.202164
3	100x	-	-0.918861	1.000000	1.000000	0.188999
4	93x	-	-1.061174	1.000000	1.000000	0.180520
5	93x	-	-1.115705	1.000000	1.000000	0.175499
6	92x	-	-1.210418	1.000000	1.000000	0.158094
7	90x	-	-1.277980	1.000000	1.000000	0.148165
8	90x	-	-1.606430	1.000000	1.000000	0.150173
9	84x	-	-1.320933	0.995483	0.868132	0.146268
10	81x	-	-1.421647	0.996387	0.874158	0.145375
11	80x	-	-1.401929	0.996886	0.845090	0.150619

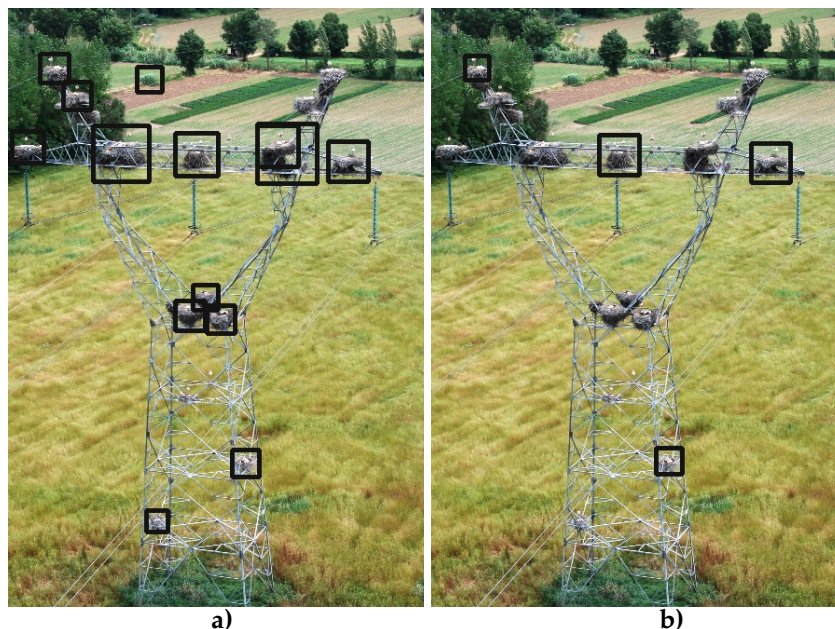
Figura 4. 14 - Primeira etapa do processo de treino do classificador.

Para a dimensão da subjanela, pretendeu-se que a escolha fosse a que permitisse a melhor relação capacidade/tempo de detecção. Uma janela de maiores dimensões (ex. 50x50) possibilitaria que o processo de treino e de detecção fosse

menos demorado. Por outro lado não seria possível identificar ninhos de dimensão inferior à da subjanela. Acresce que devido à quantidade de imagens de treino utilizadas, seria necessário que a máquina utilizada possuísse uma maior quantidade de memória RAM. Já se a subjanela fosse mais reduzida, o tempo de treino necessário escalaria, forçando um conjunto de imagens de treino inferior.

Terminado o processo de treino, foi então possível utilizar o classificador para detectar os ninhos de cegonha. Nesta fase será necessário definir dois parâmetros, que influenciarão directamente nos resultados obtidos:

- Factor de escala: Durante o processo de treino foi escolhida uma subjanela de dimensão 20x20. Para a detecção, a janela de pesquisa deverá adoptar as mesmas dimensões. No entanto, deverá ser possível identificar ninhos de maiores proporções. Deste modo existe um processo cíclico, onde em cada iteração a totalidade da imagem é analisada com recurso a uma janela de pesquisa escalada pelo factor definido. Nas figuras 4.15 são apresentados os resultados da detecção de ninhos utilizando diferentes factores de escala.



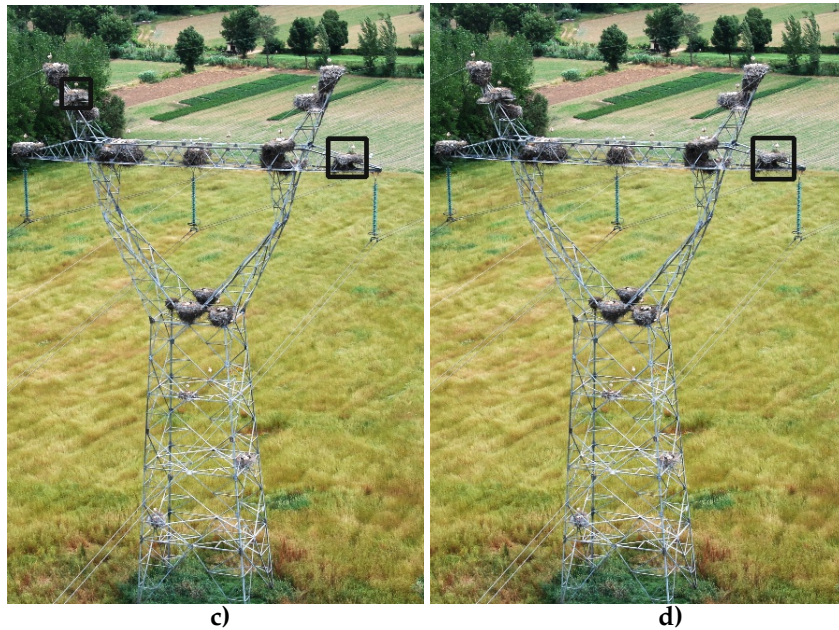


Figura 4. 15 - Resultados da detecção de ninhos de cegonha utilizando diferentes factores de escala. a) Factor de escala 1.01. b) Factor de escala 1.1. c) Factor de escala 1.2. Factor de escala 1.3.

Verifica-se que à medida que o factor de escala aumenta existe uma deterioração dos resultados. Utilizando um factor de pequena dimensão foi possível identificar 12 dos 14 ninhos existentes. Já com um escalonamento de 30% apenas um ninho foi detectado. A contrapartida de considerar o menor factor de escala reside no maior número de iterações necessárias, o que influenciará negativamente a performance do algoritmo. Apesar desta desvantagem, e tendo em conta que se pretende a melhor taxa de detecção possível, foi escolhido como factor de escala 1.01.

- *Minimum Hit*: Tipicamente, durante as iterações do processo de detecção o mesmo ninho é identificado diversas vezes. Este parâmetro permite definir o número mínimo de reconhecimentos para que uma detecção seja considerada válida. Nas figuras 4.16 são apresentados os resultados da utilização de diferentes valores para esta variável. Um *Minimum Hit* de zero significa que todas as identificações serão consideradas válidas. Este valor é o que permite a obtenção de melhores taxas de detecção, mas é também o que apresenta a maior quantidade de falsos positivos. No exemplo da figura 4.16, verifica-se que em a) apenas 1 dos ninhos existentes não foi reconhecido. Por outro lado foi o que apresentou a maior quantidade de falsos positivos, totalizando 5 identificações incor-

rectas. Nas figuras b) e c) os resultados foram semelhantes apresentando um bom equilíbrio entre detecções correctas e incorrectas. Por fim a figura d) foi a que apresentou a pior taxa de detecção e apesar de continuar a existir um ninho incorrectamente identificado, a configuração utilizada é a que possibilita a menor quantidade de falsos positivos.

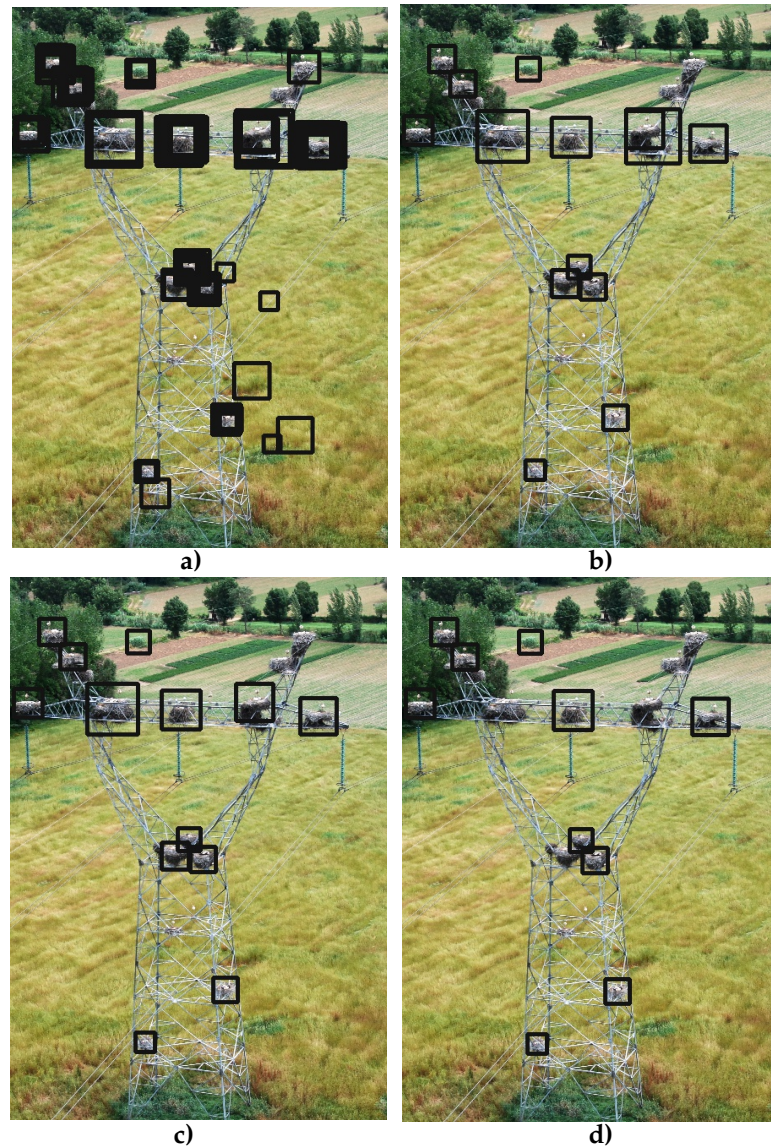


Figura 4. 16 - Resultados da detecção de ninhos utilizando diferentes valores para o parâmetro *Minimum Hit*. a) *Min Hit* = 0, b) *Min Hit* = 1, c) *Min Hit* = 2, d) *Min Hit* = 3.

No âmbito do algoritmo desenvolvido, considerou-se que um ninho apenas será considerado válido se tiver sido identificado pelo menos duas vezes.

Na figura 4.17 é apresentado um diagrama que resume as etapas do processo de detecção de ninhos de cegonha.

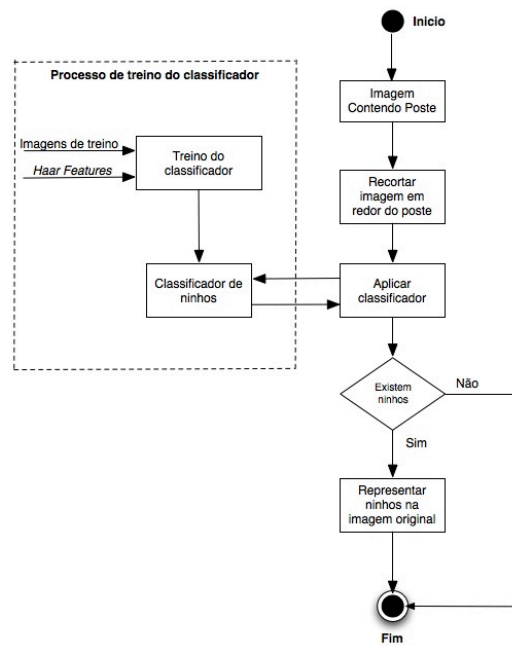


Figura 4. 17 - Diagrama do processo de detecção de ninhos de cegonha.

Nas secções anteriores foi detalhado o funcionamento do algoritmo desenvolvido. No capítulo seguinte serão apresentados os resultados dos testes efectuados.

Análise dos Resultados

Neste capítulo são apresentados os resultados dos testes efectuados. Será efectuada uma análise crítica dos mesmos, seguindo-se uma comparação com os trabalhos já existentes. Algumas das imagens utilizadas nos testes são exibidas no capítulo 8 – Anexo.

Os testes foram realizados numa máquina equipada com processador *Intel Core 2 Duo 2.4 Ghz*, 8 GB de memória RAM e sistema operativo Mac OS X.

5.1 Identificação de postes eléctricos em imagens

Para a realização dos testes foram utilizadas um total de 150 imagens, 100 contendo postes eléctricos inseridos e as restantes 50 sem qualquer poste. O primeiro conjunto de 100 imagens permite testar a capacidade do algoritmo identificar os postes em cenários variados e díspares. Já no segundo conjunto de 50 imagens pretendeu-se verificar a quantidade de postes incorrectamente identificados. Como forma de expressar os resultados serão utilizadas as seguintes métricas:

$$Precision = \frac{Postes\ Correctamente\ Identificados}{Postes\ Correctamente\ Identificados + Falsos\ Positivos}$$

$$Recall = \frac{Postes\ Correctamente\ Identificados}{Total\ de\ Postes\ Identificados}$$

$$\text{Taxa de falsos positivos} = \frac{\text{Postes Incorrectamente Identificados}}{\text{Total de Postes Identificados}}$$

O primeiro conjunto de 100 imagens possuía um total de 100 postes (um por cada imagem), dos quais 91 foram correctamente reconhecidos. Nas restantes 9 imagens, existiram 4 postes que não foram detectados e 5 situações em que a identificação foi incorrecta, que consideraremos como falsos positivos. Na figura 5.1 é apresentado um caso em que a região que delimita o poste não engloba a totalidade do mesmo. Para este facto contribuiu o caminho que se encontra na retaguarda do poste que foi considerado como sendo uma das arestas horizontais. Em relação aos postes que não foram reconhecidos, ficou-se a dever essencialmente a dois motivos:

- Complexidade da paisagem: Apesar dos métodos utilizados para reduzir o impacto da paisagem no processo de detecção, em cenários mais complexos poderá existir alguma dificuldade em identificar a estrutura do poste.
- Ângulo de visualização não contemplado: Uma das ressalvas do algoritmo reside no ângulo de visualização do poste, considerando-se que o mesmo se encontra em posição frontal. Por posição frontal entende-se que as arestas respeitam as condições expressas no capítulo anterior. Quando tal não se verifica, as probabilidades do poste ser identificado são menores.

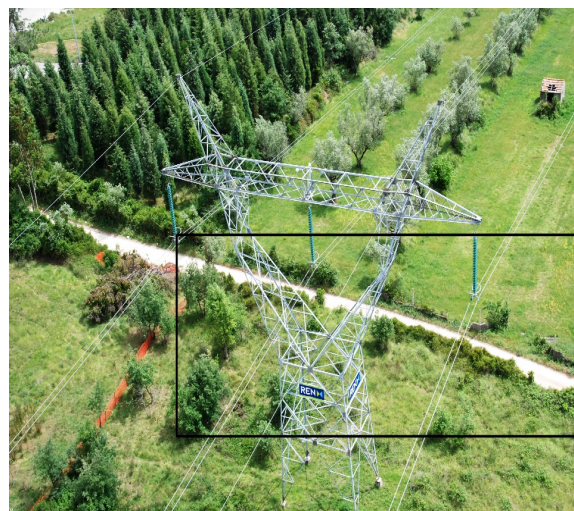


Figura 5.1 - Incorrecta identificação de um poste.

No segundo conjunto de imagens que não continham postes, de entre as 50 existentes foram identificados incorrectamente 2 postes. Em ambas as situações, o erro deveu-se a condições particulares da paisagem, levando à extracção de arestas que respeitavam a condição de reconhecimento de um poste.

Em relação ao tempo necessário para identificar um poste, em média a análise de uma imagem demora cerca de 4 segundos.

Na tabela 5.1 são apresentados os resultados da identificação de postes eléctricos do algoritmo, assim como os obtidos pelos protótipos existentes.

Tabela 5. 1 – Comparação dos resultados obtidos com os protótipos existentes

	<i>Precision</i>	<i>Recall</i>	Taxa de FP
Algoritmo proposto	92,8%	91%	5,2%
Liu [Liu, W., 2010]	70%	70%	-
João Fernandes (imagens) [Fernandes, J., 2012]	76,3%	78,3%	9%

Comparando os resultados verifica-se que o algoritmo proposto apresenta uma taxa de detecção superior aos protótipos existentes, assim como uma taxa de identificações incorrectas inferior. De notar que os resultados de João Fernandes são referentes aos testes em imagens singulares e como tal sem os benefícios da correlação temporal entre *frames*. Mais, considerou-se para o cálculo da taxa de falsos positivos todos os postes reconhecidos com região delimitante incorrecta. Última nota para referir que as imagens utilizadas nos testes dos três algoritmos são diferentes, pelo que este aspecto deverá ser tido em consideração.

Como mencionado no capítulo 3, existem alguns trabalhos dedicados à detecção de postes eléctricos. Com a devida ressalva que modelo de poste utilizado é diferente, em [Golightly, I., Jones., D., 2003] foram identificados em média 86,7% dos postes existentes, com uma taxa de falsos positivos média de 22%. Já em [Tilawat, J., *et al*, 2010] pretendia-se detectar o mesmo tipo de poste desta tese, obtendo uma taxa de identificação de 97%. No entanto este valor é alcançado à custa de uma taxa de falsos positivos de cerca de 40%.

5.2 Identificação de postes eléctricos em vídeos.

O foco dos testes foi realizado com recurso a imagens, uma vez que permite obter uma amostra mais abrangente de cenários. Adicionalmente, devido aos elevados custos de processamento, a obtenção de resultados em imagens é menos demorada que em vídeos. No entanto e como prova do conceito, foram também efectuados testes com recursos a 3 ficheiros vídeo de curta duração (inferior a 5 min). As amostras utilizadas foram obtidas a partir de duas fontes. A primeira com formato *.mov* possuía uma *frame-rate* de 29 e uma resolução de 1440x1080. A segunda encontrava-se codificada com o *codec* AVCHD, possuindo a mesma resolução que a anterior fonte, mas com uma *frame-rate* de 25. Em ambas os vídeos, os postes encontravam-se inseridos em ambientes florestais/agrícolas. Os resultados obtidos são apresentados na tabela 5.2.

Tabela 5. 2 - Resultados da identificação de postes eléctricos em vídeos.

	<i>Vídeo 1</i>	<i>Vídeo 2</i>	<i>Vídeo 3</i>	Total
Postes Existentes	11	12	17	40
Postes Identificados	11	12 (+1)	17	40 (+1)
<i>Precision</i>	100%	92,3	100%	97,6%
<i>Recall</i>	100%	100%	100%	100%
Taxa de FP	0%	7,7%	0%	2,4%

Comparando os resultados da análise de imagem e de vídeo, verifica-se que a taxa de detecção aumentou de 91% (imagens) para 100% (vídeo). Dever-se-á ressaltar a reduzida amostra de dados utilizada neste último teste, o que poderá ter influenciado os resultados. Não obstante, era expectável que a percentagem de poste identificados aumentasse uma vez que existiria um maior número de amostras para cada poste. Adicionalmente, a quantidade de falsos positivos diminuiu, uma vez que na análise de ficheiros vídeo existe a obrigatoriedade de reconhecimento de um poste em duas *frames* sucessivas. Ainda assim no *vídeo 2* existiu um poste incorrectamente identificado.

Recordamos que no trabalho de João Fernandes, utilizando a correlação temporal entre *frames*, foi obtida uma taxa de detecção de 80% para os postes eléctricos. O conjunto de vídeos utilizado foi consideravelmente superior e exis-

tiram algumas amostras que poderão ter influenciado negativamente os resultados.

5.3 Comparação de resultados com outros trabalhos

No capítulo 3 foram apresentados alguns trabalhos que utilizavam técnicas semelhantes para a identificação de objectos que não postes eléctricos. Em [Fields, A., Green, R., 2012] o algoritmo de detecção de veículos tinha obtido uma taxa de detecção de 100%. Já em [Hocenski, Z., *et al*, 2006] foram identificadas 98% das imperfeições presentes em azulejos. Considerando que o algoritmo proposto foi capaz de reconhecer 91% de postes em imagens e 100% em vídeos, fica demonstrado que a utilização do *Canny edge detector* e a transformação de *Hough* são meios eficientes no reconhecimento de objectos.

5.4 Identificação de ninhos de cegonha em imagens

As métricas utilizadas para a detecção dos ninhos de cegonha são semelhantes às usadas na identificação dos postes eléctricos.

$$Precision = \frac{Ninhos\ Correctamente\ Identificados}{Ninhos\ Correctamente\ Identificados + Falsos\ Positivos}$$

$$Recall = \frac{Ninhos\ Correctamente\ Identificados}{Total\ de\ Ninhos\ Identificados}$$

$$Taxa\ de\ falsos\ positivos = \frac{Ninhos\ Incorrectamente\ Identificados}{Total\ de\ Ninhos\ Identificados}$$

Nos testes efectuados para a identificação de postes, de entre os 100 existentes, 91 foram correctamente reconhecidos. Destes, 53 continham pelo menos um ninho e os restantes 38 não possuíam nenhum. A distribuição da quantidade de ninhos existentes por postes encontra-se presente na tabela 5.3.

Tabela 5. 3 - Quantidade de ninhos existentes nos postes identificados.

	1 ninho	2 ninhos	3 a 5 ninhos	Mais de 5 ninhos
Postes	19	8	11	15

Os postes identificados continham um total de 336 ninhos. Destaque para os 15 postes com mais de 5 ninhos, chegando a existir um com 25 ninhos.

Os resultados da detecção de ninhos são apresentados na figura 5.2. Foram efectuadas 312 identificações das quais 273 correspondiam a uma correcta detecção e os restantes 39 eram falsos positivos.

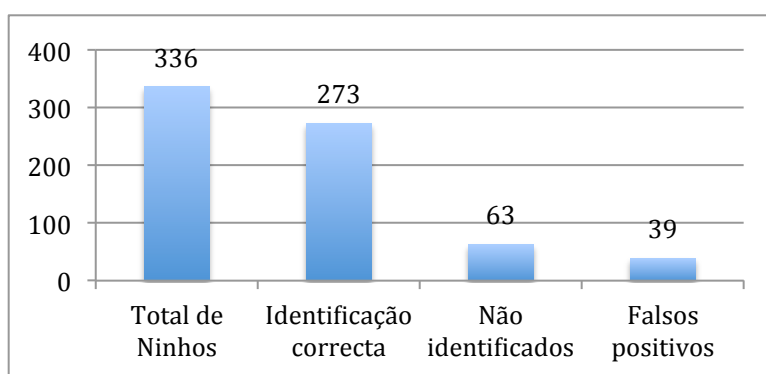


Figura 5. 2 - Resultados da detecção de ninhos de cegonha em imagens.

De entre as identificações incorrectas, verificaram-se duas situações:

- Alguns objectos presentes na paisagem foram reconhecidos como sendo ninhos. Não existindo uma relação aparente entre os mesmos, foram identificados como sendo ninhos, árvores, arbustos, sinais de trânsito e secções de algumas habitações. Na figura 5.3 é apresentada uma situação em que um arbusto foi identificado como sendo um ninho.
- Sectores do poste foram reconhecidos como contendo ninhos. Nas imagens utilizadas para o treino do classificador, para além dos ninhos existiam ainda pequenas secções do poste. É possível que algumas destas características do poste tenham sido consideradas no processo de treino e verificando-se determinadas condições na paisagem, se considere incorrectamente a existência de um ninho. Na figura 5.4 uma secção inferior do poste foi identificada como contendo um ninho.



Figura 5. 3 - Incorrecta identificação de um ninho contido na paisagem.

Nos testes efectuados verificou-se que 18,8% dos ninhos não foram detectados. Não foi encontrado nenhum padrão comum para estes casos, mas verificou-se uma maior dificuldade de identificação dos ninhos existentes na secção inferior do poste e inseridos 'dentro' do poste. Para a primeira situação, uma possível justificação poderá ser a diferença dos ângulos de visualização dos ninhos nas secções superiores para os das secções inferiores do poste. Apesar de se ter tentado incluir no treino do classificador ninhos visualizados de diferentes posições, a maioria encontrava-se nas regiões superiores do poste e como tal apresentam uma forma ligeiramente diferente dos que se encontram nas regiões inferiores. Já para a segunda situação mencionada, alguns dos ninhos não detectados encontravam-se parcialmente encobertos por arestas do poste, o que poderá ter impossibilitado a sua identificação. No entanto existiram ninhos nas condições anteriores que foram correctamente reconhecidos.

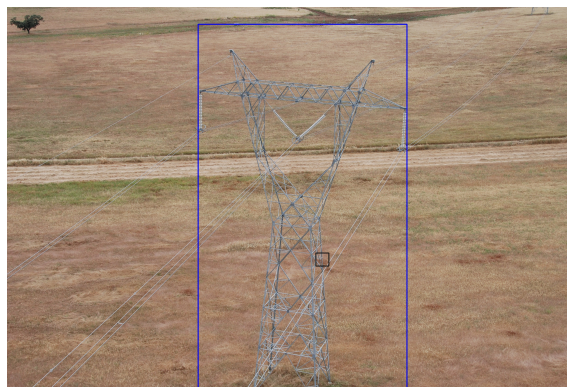


Tabela 5. 4 - Incorrecta identificação de uma secção do poste como contendo um ninho.

Última nota para os tempos de necessário para a detecção de ninhos. Nas configurações actuais, o processamento de uma imagem demora em média cerca de 6 segundos.

Na tabela 5.4 são apresentados os resultados da detecção de ninhos pelo algoritmo proposto, assim como dos protótipos existentes.

Tabela 5. 5 - Comparação dos resultados dos protótipos existentes na detecção de ninhos em imagens.

	<i>Precision</i>	<i>Recall</i>	Taxa de FP
Algoritmo proposto	87,5%	81,3%	12,5%
Liu [Liu, W., 2010]	57,9%	61%	7,7%
João Fernandes (imagens) [Fernandes, J., 2012]	77,3%	52,6%	29,4%

Verifica-se que o algoritmo proposto apresenta uma taxa de detecção superior a ambos os trabalhos existentes. O algoritmo de Liu foi o que apresentou a taxa de falsos positivos mais reduzida, correspondendo a uma única identificação incorrecta. No entanto o conjunto de dados para testes foi relativamente reduzido, sendo que os resultados não são totalmente conclusivos. Os dados pertencentes ao trabalho de João Fernandes são referentes a imagens singulares, o que poderá justificar a reduzida capacidade de detecção.

5.5 Identificação de ninhos de cegonha em vídeos

Os vídeos empregues nestes testes são os mesmos que previamente tinham sido utilizados na identificação de postes eléctricos. Os resultados são apresentados na tabela 5.5.

Novamente, devido às dimensões dos dados de teste, os resultados apresentados não são totalmente conclusivos. No entanto demonstram uma ligeira melhoria nas taxas de detecção e no número de identificações incorrectas. Verifica-se a existência de um falso positivo no *vídeo* 3. Isto ocorreu devido a dois reconhecimentos de ninhos incorrectos em *frames* sucessivas. Apesar das detecções terem ocorrido em locais distintos, são consideradas como sendo as mesmas, uma vez que não é efectuado o relacionamento dos resultados obtidos entre *frames*.

Tabela 5. 6 - Resultados da detecção de ninhos de cegonha em ficheiros de vídeo.

	<i>Vídeo 1</i>	<i>Vídeo 2</i>	<i>Vídeo 3</i>	Total
Ninhos Existentes	0	2	8	10
Ninhos Identificados	0	2	7 (+1)	9(+1)
<i>Precision</i>	-	100%	87,5%	90%
<i>Recall</i>	-	100%	87,5%	90%
Taxa de FP	0%	0%	12,5%	10%

Comparativamente com os resultados obtidos pelo algoritmo de João Fernandes na análise de vídeos, os aqui apresentados são ligeiramente superiores. Recordo que utilizando a correlação temporal de *frames* havia sido obtido uma taxa de detecção de 78% e uma percentagem de identificações incorrectas de 22%.

5.6 Comparação de resultados com outros trabalhos

Como demonstrado no terceiro capítulo existem diversos trabalhos que utilizam as *haar features* como meio de detectar objectos.

No algoritmo original de detecção de faces de Viola e Jones foi possível obter uma taxa de detecção de 90% e de identificações incorrectas de 9%. Um possível motivo para a melhor capacidade de reconhecimento em comparação com o algoritmo aqui proposto, poderá residir no maior número de etapas utilizadas no treino do classificador. Já a possível menor complexidade dos cenários (*background*) poderá ter contribuído para o reduzido número de falsos positivos.

Tabela 5. 7 - Resultados dos diferentes trabalhos que utilizam *haar features*.

	Ke e Kang*	Yong <i>et al</i>	Han* <i>et al</i>	Hota* <i>et al</i>	Monteiro <i>et al</i>
Taxa detecção	81% - 100%	91%	76% - 91%	60% - 97%	90%
Taxa de FP	-	2%	-	21% - 120%	7%

(*) Resultados variam consoante os testes efectuados.

Na tabela 5.6 são resumidos os resultados dos restantes trabalhos. Para alguns destes é apresentado um intervalo de valores, dependendo das situações em que os testes foram realizados. Quando comparados os resultados da capacidade de identificação com os do algoritmo proposto, verifica-se que estes se encontram dentro da média apresentada. De notar que os ninhos têm uma forma difusa, razão pela qual a sua detecção se revela algo problemática.

Conclusão e Trabalho Futuro

O objectivo desta tese consistia na identificação de ninhos de cegonha em postes eléctricos. Como tal, pretendia-se desenvolver uma aplicação capaz de automaticamente identificar ninhos de cegonha em postes eléctricos com recurso a imagens/vídeos capturados por um helicóptero. Uma correcta detecção possibilitaria uma redução de custos de inspecção, eliminando-se a necessidade de vistorias dedicadas à monitorização de cegonhas. Indirectamente, através duma gestão adequada, poderia permitir uma redução nos incidentes causados por cegonhas, diminuindo desta forma as falhas na distribuição eléctrica.

Deste modo foi desenvolvida uma aplicação que no essencial se centrava nos seguintes aspectos:

- Identificação de postes eléctricos. Este processo é efectuado com recurso ao *Canny edge detector* e transformada de *Hough*.
- Detecção de ninhos de cegonha. Este procedimento consistia na utilização de um classificador de ninhos previamente treinado.

Com base nos resultados obtidos, verifica-se que o algoritmo proposto é capaz de correctamente identificar 91% dos postes eléctricos e 81% dos ninhos de cegonha. Quando comparado com os protótipos existentes verifica-se uma melhoria global dos resultados. . No entanto, com base nestes mesmos resultados, a capacidade de detecção de ninhos não dispensa completamente a actual verificação manual. Existem algumas metodologias que poderão ser empregues no futuro:

- Utilização de um segundo classificador de ninhos. Como demonstrado em [Hota, R., *et al*, 2010], a combinação de dois classificadores possibilita a obtenção de melhores resultados.
- Relacionamento das detecções efectuadas em *frames* sucessivas. Ficou provado em [Fernandes, J., 2012] que a correlação temporal entre *frames* apresenta melhores índices de detecção quando comparado com o processamento individual de imagens.

Estas técnicas a serem aplicadas permitiriam uma melhor capacidade de reconhecimento, diminuindo igualmente o número de identificações incorrectas.

Em relação ao processo de identificação de postes eléctricos, apesar dos resultados serem satisfatórios, podem no futuro ser corroborados com coordenadas *GPS*. Actualmente existe a possibilidade de associar as coordenadas do helicóptero aos ficheiros de vídeo/imagem nos instantes em que são capturados. Relacionando essa informação com uma base de dados existente contendo as coordenadas de todos os postes, permitiria que os resultados da identificação de um poste eléctrico fossem fiáveis.

Existe ainda um problema ao nível da performance do algoritmo. Actualmente os custos de processamento de uma imagem são elevados, demorando a sua análise vários segundos. A abordagem tomada nesta tese consistia na obtenção dos melhores resultados possíveis, relegando para segundo plano a performance. Mas se algumas das técnicas mencionadas neste capítulo fossem empregues seria possível adaptar os parâmetros de configuração do algoritmo.

Os objectivos desta tese foram atingidos. No entanto, existe ainda um conjunto de melhoramentos que deverão ser aplicados de modo a construir um sistema ainda mais competente, conforme discutido.



Bibliografia

[Albatroz, 2012] Albatroz Engenharia, "*Power Line Inspection Maintenance Inspection - PLMI*", http://www.albatroz-eng.com/solutions/power_line_maintenance_inspection.html,

[Buades, A., et al 2005] Buades, A., Coll, B., Morel, J., "*A review of image denoising algorithms, with a new one*", Society for Industrial and Applied Mathematics vol. 4, 2005.

[Cai, L., et al, 2010] Cai, L., Zhang, C., Li, B., Zong, C., Wang, Q., "*Traffic Sign Detection Using Surround Suppression of Textures Edges and Cascade Detector*", Annual Conference on Intelligent Transportation Systems, 2010.

[Canny, J., 1986] Canny, J. "*A Computational Approach to Edge Detection*". IEEE Transactions on pattern analysis and machine intelligence, vol. Pami-8 NO. 6, 1986.

[Cheng, W., Song, Z., 2008] Cheng, W., Song, Z., "*Power Pole Detection Based on Graph Cut*", Congress on Image and Signal Processing, 2008.

[Cooper, J., et al, 1991] Cooper, J., Venkatesh, S., Kitchen, L., "*Early Jump-out Corner Detectors*", Computer Vision and Pattern Recognition, 1991.

[Coupe, S., 2009] Coupe, S., "*The Current Status of Pairwise Geometric Histograms*", Literature Summary for PhD thesis, 2009.

[Crow, F., 1984] Crow, F., "*Summed-Area Tables for Texture Mapping*", SIGGRAPH' 84 volume 18 number 3, 1984.

[Daigavane, P., Bajaj, P., 2010] Daigavane, P., Bajaj, P., "*Road Lane Detection with Improved Canny Edge Using Ant Colony Optimization*", Third International Conference on Emerging Trends in Engineering and Technology, 2010.

- [Derpanis, K., 2007] Derpanis, K., *"Integral image-based representations"*, Department of Computer Science and Engineering York University, 2007.
- [Duda, R., Hart, P., 1972] Duda, R., Hart, P., *"Use of Hough Transformation to detect lines and curves in images"*, Communications of the ACM, New York, Janeiro 1972.
- [Fan, X., et al, 2012] Fan, X., Zhang, F., Wang, H., Lu, X., *"The System of Face Detection Based on OpenCV"*, 24th Chinese Control and Decision Conference, 2012.
- [Fernandes, J., 2012] Fernandes, J., *"Automatic Detection of Stork Nests on Very-High Voltage Towers"*, Instituto Superior Técnico – Universidade Técnica de Lisboa, 2012.
- [Fields, A., Green, R., 2012] Fields, A., Green, R., *"Characteristic-Based Vehicle Recognition for collision Detection"*, Image and Vision Computing New Zealand, 2012.
- [Freund, Y., et al. 1995] Freund, Y. e Schapire, R.E. *"A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting"*. In Computational Learning Theory: Eurocolt '95, Springer-Verlag, 1995.
- [Glasbey, C., Horgan, G., 1995] Glasbey, C., Horgan, G., *"Image Analysis for the biological Sciences"* Chapter 3 - Filters, Wiley, 1995.
- [Gokul, S., et al, 2012] Gokul, S., Kumar, G., Sreeraj, M., *"Real Time Recognition of Pedestrians and Vehicles from Videos"*, Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology, New York, 2012.
- [Golightly, I., Jones, D., 2003] Golightly, I., Jones, D., *"Corner Detection and Matching for Visual Tracking During Power Line Inspection"*, Image and Vision Computing 21, Elsevier, 2003.
- [Gunn, S., 1998] Gunn, S., *"Support vector machine for classification and regression"*, Technical Report, 1998.
- [Han, S., et al, 2009] Han, S., Han, Y., Hahn, H., *"Vehicle Detection Method using Haar-like Feature on Real Time System"*, World Academy of Science, Engineering and Technology 35, 2009.
- [Harris, A., 2003] Harris, A., *"Chapter 6: Vector Graphics"*, http://wally.cs.iupui.edu/n351/vector/Vector_Graphics.html, N351 Multimedia Videos, Indiana University-Purdue University Indianapolis, 2003.

- [Hocenski, Z., et al, 2006] Hocenski, Z., Vasilic, S., Hocenski, V., "*Improved Canny Edge Detector in Ceramic Tiles Defect Detection*", IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference, Paris, 2006.
- [Hota, R., et al, 2010] Hota, R., Jonna, K., Krishna, P., "*On-Road Vehicle detection by Cascaded Classifiers*", Compute '10 Proceedings of the third Annual ACM Bangalore Conference, 2010.
- [Jia, X., 2010] Jia, X., "*Fabric Defect Detection Based on Open Source Computer Vision Library OpenCV*", 2nd International Conference on Signal Processing Systems, 2010.
- [Ke, L., Kang, J., 2010] Ke, L., Kang, J., "*Eye Location Method Based on Haar Features*", 3rd International Congress on Image and Signal Processing, 2010.
- [Lienhart, R., Maydt, J., 2002] Lienhart, R., Maydt, J., "*An Extended Set of Haar-like Features for Rapid Object Detection*", IEEE Image Processing, 2002.
- [Liu, W., 2010] Liu, W., "*Nest and Platform detection from Video Sequences*", Albartroz Engenharia, 2010.
- [Liu, Y., et al, 2012] Liu, Y., Mejias, L., Li, Z., "*Fast Power Line Detection and Localization using Steerable Filter for Active UAV Guidance*", International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Melbourne, 2012.
- [Lu, D., Chen, C., 2008] Lu, D., Chen, C., "*Edge Detection Improvement by ant colony optimization*", Pattern Recognition Letters vol. 29, 2008.
- [Maini, R., Aggarwal, H., 2009] Maini, R., Aggarwal, H. "*Study and Comparison of Various Image Edge Detection Techniques*". International Journal of Image Processing. 2009.
- [Meynet, J., 2003] Meynet, J., "*Fast face detection using adaboost*", Signal Processing Institute, EPFL, 2003.
- [Monteiro, G., et al, 2006] Monteiro, G., Peixoto, P., Nunes, U., "*Vision-Based Pedestrian Detection using Haar-like features*", Instituto de Sistemas e Robótica da Universidade de Coimbra, 2006.
- [Mousa, A., 2012] Mousa, A., "*Canny Edge-Detection Based Vehicle Plate Recognition*", International Journal of Signal Processing, Image Processing and Pattern Recognition, 2012.
- [Papageorgiou, C., et al., 1998] Papageorgiou, C., Oren, M., Poggio, T., "*A General Framework for object detection*", Sixth International Conference on Computer Vision, 1998.

[Qu, Y., et al., 2005] Qu, Y., Li, C., Zheng, N., “*Airport Detection Base on Support Vector Machine from A Single Image*”, Information Communications and Signal Processing, 2005.

[Rai, H., et al, 2011] Rai, H., Jonna, K., Krishna, P., “*Video Analytics Solution for Tracking Customer Locations in Retail Shopping Malls*”, KDD’11 Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2011.

[REN, 2003] REN, “*REN em linha com o desenvolvimento sustentável*”, <http://www.centrodeinformacao.ren.pt/PT/publicacoes/PublicacoesGerais/REN%20em%20linha%20com%20o%20desenvolvimento%20sustentável.pdf>, Último acesso 10 Novembro 2012.

[REN, 2012] REN, “*REN, Relatório & Contas 2012*”, http://relatorioecontas2011.ren.pt/media/172099/sustentabilidade_ren.pdf, Último acesso 12 Novembro 2012.

[Rosa, G., et al, 2005] Rosa, G., Encarnação, V., Candelária, M., “*V Censo Nacional de Cegonha-branca Ciconia ciconia (2004)*”, Sociedade Portuguesa para o estudo de aves, Instituto da Conservação da Natureza, 2005.

[Sarkar, S., Boyer, K., 1991] Sarkar, S., Boyer, K., “*On Optimal Infinite Impulse Response Edge Detection Filters*”, Transactions on Pattern Analysis and Machine Intelligence vol.13, 1991.

[Santos, F., et al, 2002] Santos, F., Forbes, K., Moita, R., “*Climate Change in Portugal. Scenarios, Impacts and Adaptation Measures - SIAM Project*”, Gradiva, Portugal, 2002.

[Soltany, M., et al, 2011] Soltany, M., Zadeh, S., Pourreza, H., “*Fast and Accurate Pupil Positioning Algorithm using Circular Hough Transform and Gray Projection*”, 2011 International Conference on Computer Communication and Management, Singapore, 2011.

[Tilawat, J., et al, 2010] Tilawat, J., Auephanwiriyaikul, S., Theera-Umpon, N., “*Automatic Detection of Electricity Pylons in Areal Video Sequences*”, International Conference on Electronics and Information Engineering, 2010.

[Viola, P., Jones, M., 2001] Viola, P., Jones, M., “*Rapid Object Detection using a Booted Cascade of Simple Features*”. CVPR ’01. 2001.

[Wafi, Z., et al, 2010] Wafi, Z., Ahmad, R., Paulraj, P., “*Highways Traffic Surveillance System (HTSS) using OpenCV*”, Control and System Graduate Research

Colloquium, 2010.

[Wang, B., *et al*, 2010] Wang, B., Chen, X., Wang, Q., Liu, L., Zhang, H., Li, B., *"Power Line Inspection with a Flying Robot"*, 1st International Conference on Applied Robotics for the Power Industry, Canada, 2010.

[Yong, X., *et al*, 2011] Yong, X., Zhang, L., Song, Z, Hu, Y., Zheng, L., Zhang, J., *"Real-Time Vehicle Detection Based on Haar Features and Pairwise Geometrical Histograms"*, International Conference on Information and Automation, China, 2011.

Anexo

Neste capítulo são apresentados alguns dos resultados obtidos nos testes efectuados. Pretende-se deste modo ilustrar o conjunto de situações variadas aos quais o algoritmo se encontrou exposto.



Figura 8. 1 - Poste Correctamente identificado



Figura 8. 2 - Incorrecta identificação de um poste.

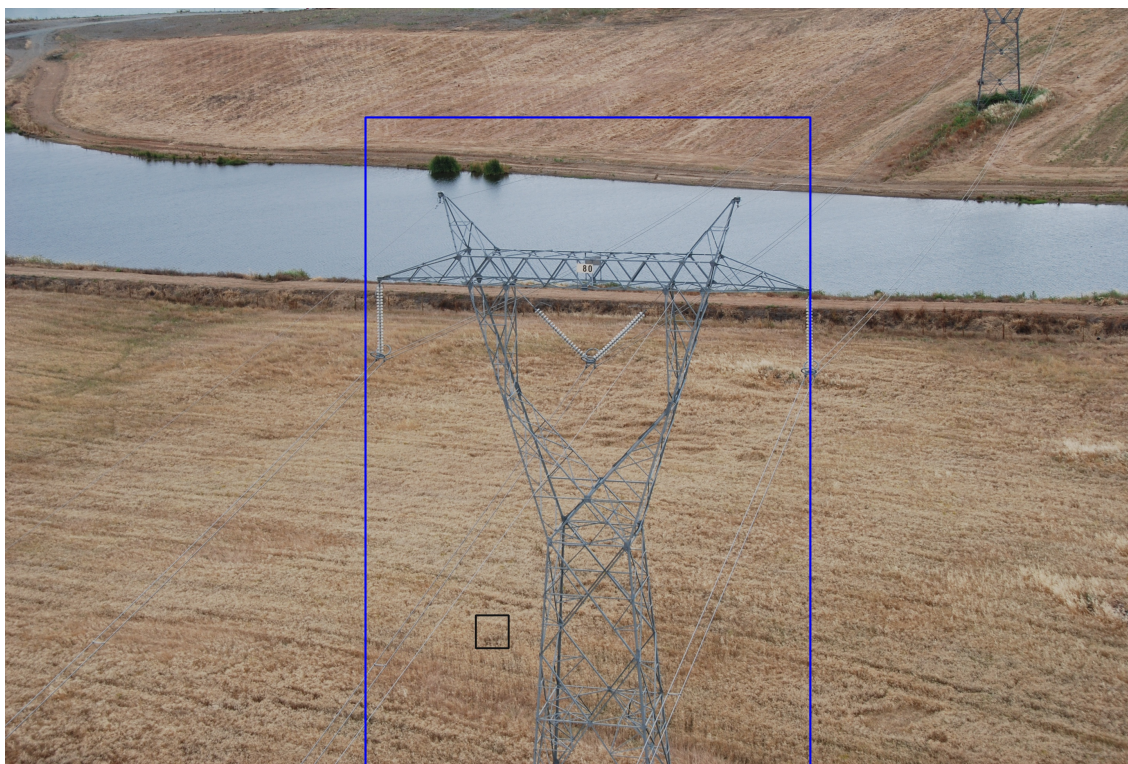


Figura 8. 3 - Identificação de um ninho na paisagem (falso positivo).



Figura 8. 4 - Incorrecta identificação de um ninho no poste (falso positivo).

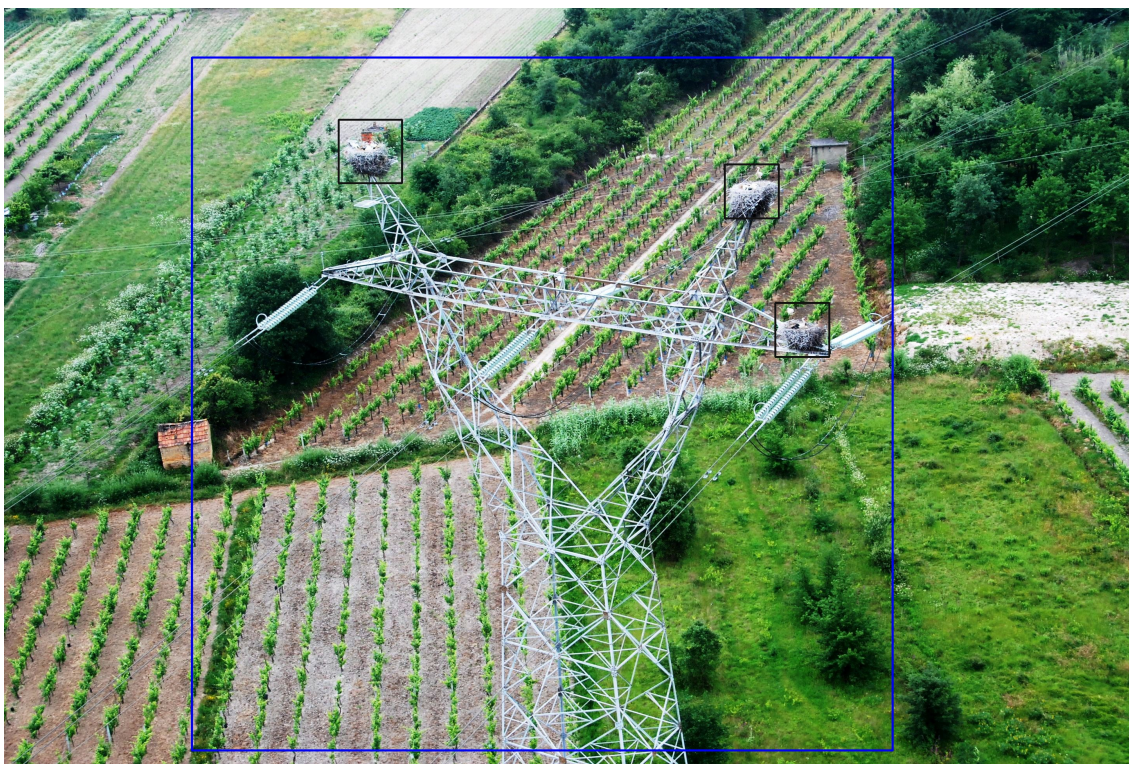


Figura 8. 5 - Identificação da totalidade dos ninhos existentes no poste.



Figura 8. 6 - Não detecção de um ninho, aliado a duas identificações incorrectas



Figura 8. 7 - Identificação de um poste em posição não frontal.

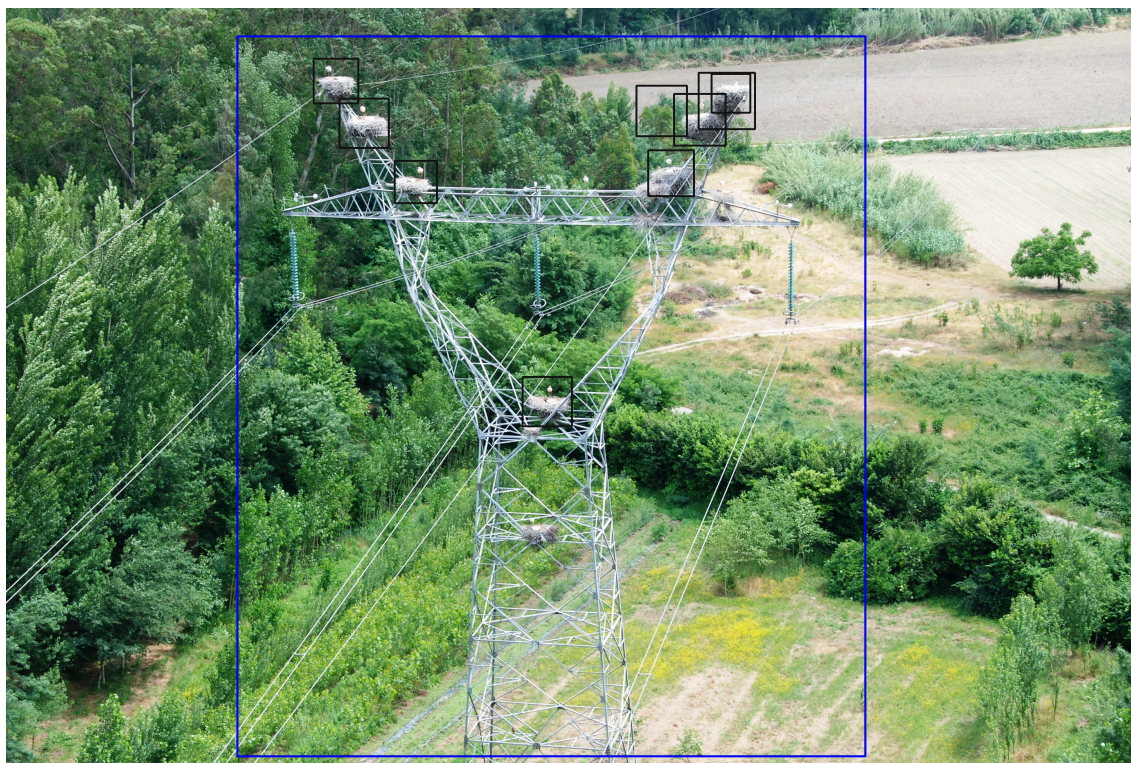


Figura 8. 8 - Identificação de 7 dos 8 ninhos existentes.

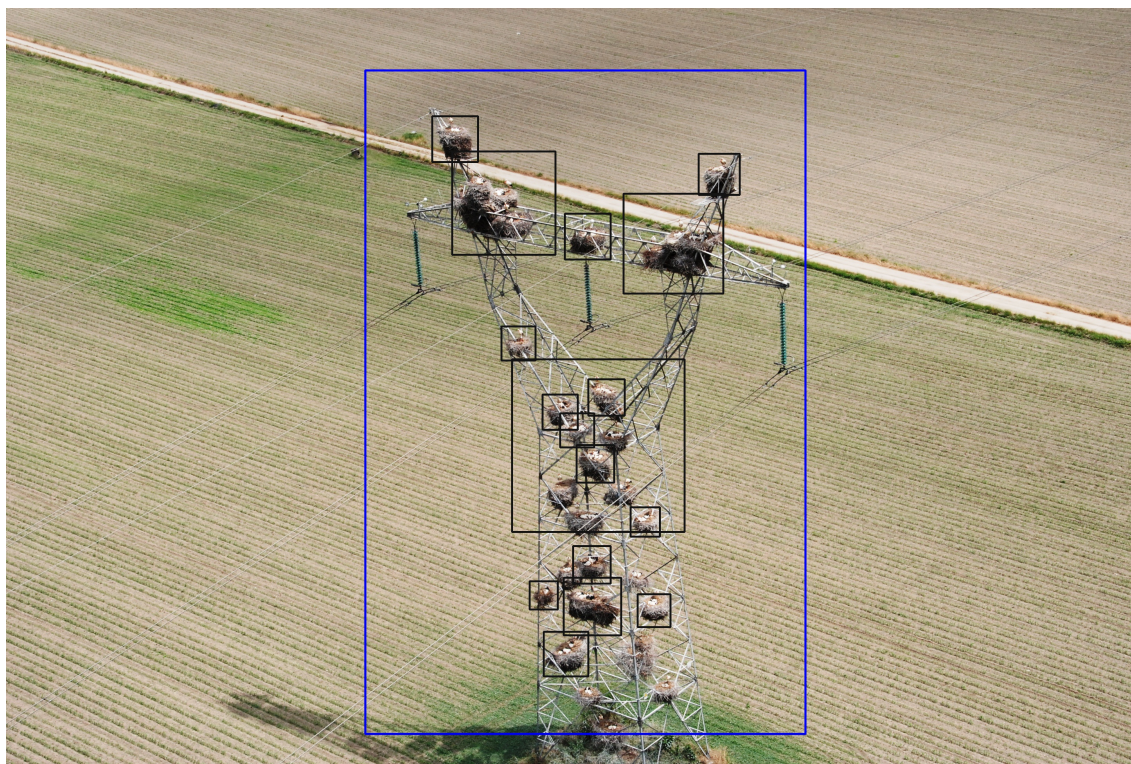


Figura 8. 9 - Exemplo de um poste contendo um grande número de ninhos (25).



Figura 8. 10 - Sequência de análise das *frames* extraídas de um ficheiro vídeo.